

FIGURE 1A

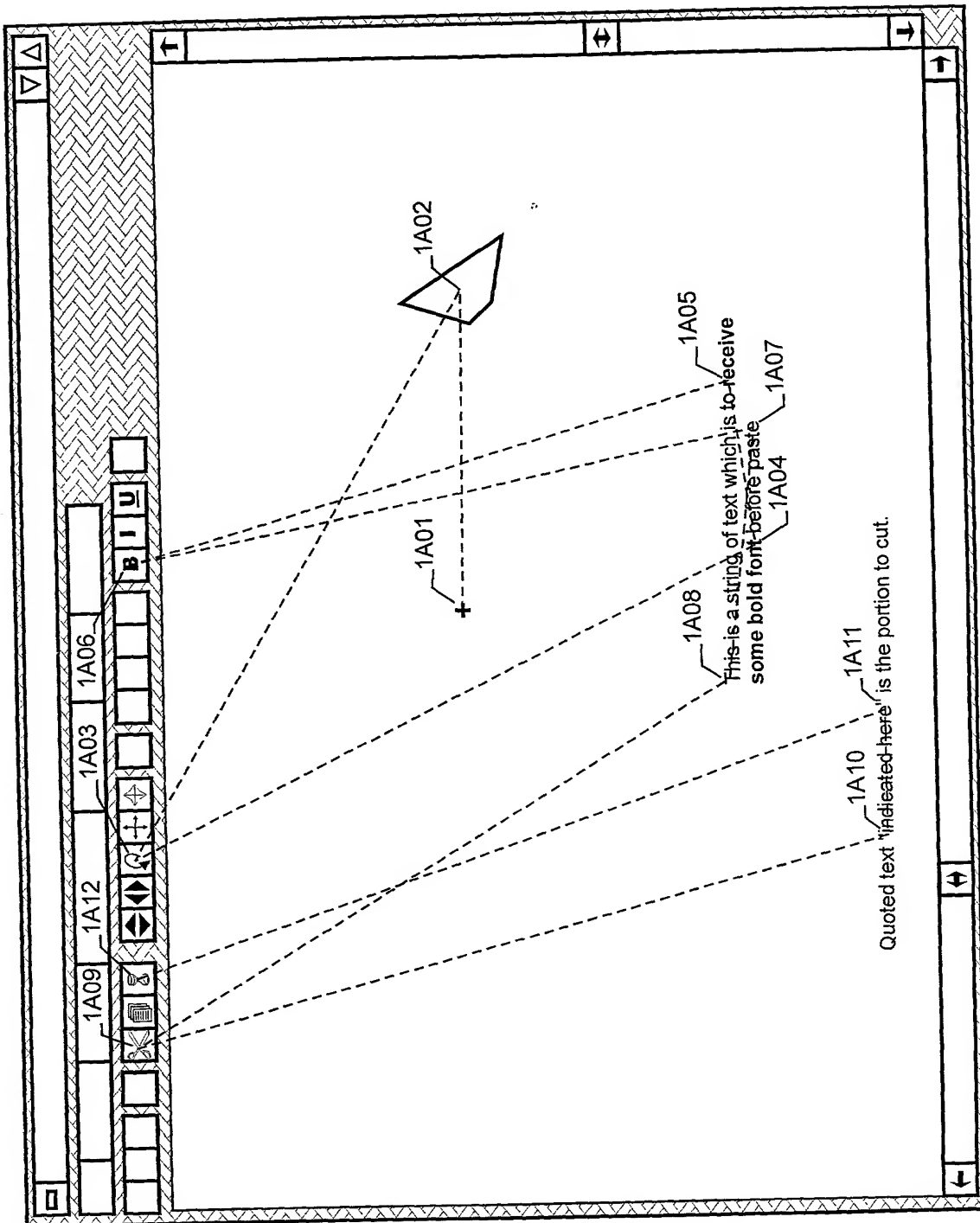


FIGURE 1A

FIG. 1B

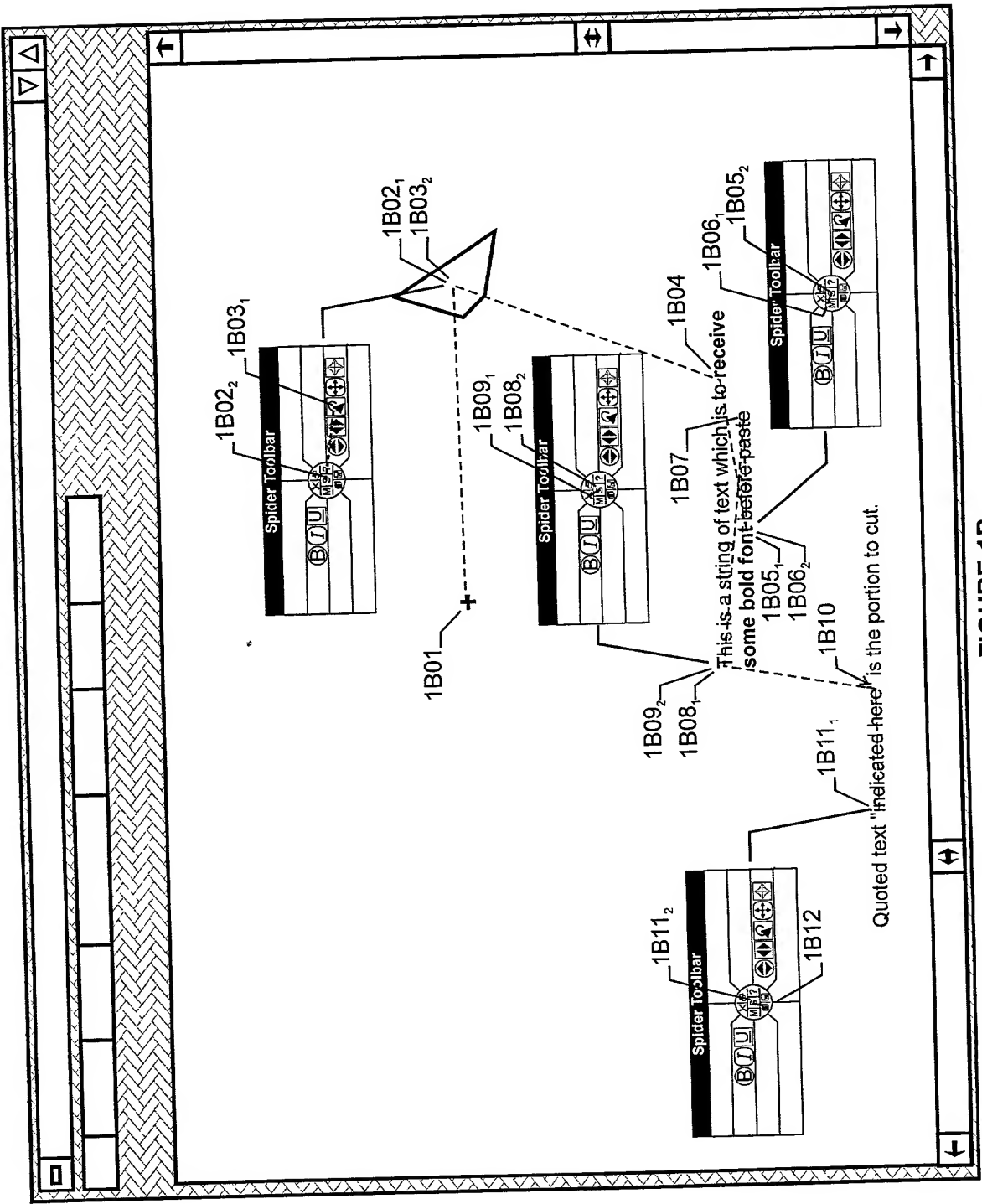


FIGURE 1B

Traditional ToolBar Analysis

Traverse	Start		End		Size		Physical Effort		
	X	Y	X	Y	D	W	Per Traverse	Common	Toolbar Specific
1A01-1A02	5.56	4.76	8.20	4.71	2.64	0.40	2.72	2.72	0.00
1A02-1A03	8.20	4.71	3.57	7.37	5.34	0.25	4.42	0.00	4.42
1A03-1A04	3.57	7.37	6.08	2.43	5.54	0.08	6.11	0.00	6.11
1A04-1A05	6.08	2.43	7.41	2.61	1.34	0.08	4.07	4.07	0.00
1A05-1A06	7.41	2.61	6.00	7.40	4.99	0.25	4.32	0.00	4.32
1A06-1A07	6.00	7.40	7.03	2.43	5.08	0.08	5.99	0.00	5.99
1A07-1A08	7.03	2.43	4.96	2.61	2.08	0.08	4.70	4.70	0.00
1A08-1A09	4.96	2.61	2.00	7.40	5.63	0.25	4.49	0.00	4.49
1A09-1A10	2.00	7.40	3.68	1.44	6.19	0.08	6.27	0.00	6.27
1A10-1A11	3.68	1.44	4.68	1.44	1.00	0.08	3.64	3.64	0.00
1A11-1A12	4.68	1.44	2.60	7.40	6.31	0.25	4.66	0.00	4.66
TOTALS					46.15		51.40	15.13	36.26

Spider ToolBar Analysis

Traverse	Start		End		Size		Physical Effort		
	Start.X	Start.Y	End.X	End.Y	D	W	Per Traverse	Common	Toolbar Specific
1B01-1B02 ₁	5.56	4.76	8.20	4.71	2.64	0.40	2.72	2.72	0.00
1B02 ₂ -1B03 ₁	6.59	5.78	7.21	5.69	0.63	0.15	2.06	0.00	2.06
1B03 ₂ -1B04	8.20	4.71	7.41	2.61	2.24	0.08	4.81	0.00	4.81
1B04-1B05 ₁	7.41	2.61	6.08	2.43	1.34	0.08	4.07	4.07	0.00
1B05 ₂ -1B06 ₁	8.05	1.69	7.86	1.76	0.20	0.15	0.43	0.00	0.43
1B06 ₂ -1B07	6.08	2.43	7.03	2.43	0.95	0.08	3.57	0.00	3.57
1B07-1B08 ₁	7.03	2.43	4.96	2.61	2.08	0.08	4.70	4.70	0.00
1B08 ₂ -1B09 ₁	6.45	3.59	6.40	3.68	0.10	0.09	0.19	0.00	0.19
1B09 ₂ -1B10	4.96	2.61	4.68	1.44	1.20	0.08	3.91	0.00	3.91
1B10-1B11 ₁	4.68	1.44	3.68	1.44	1.00	0.08	3.64	3.64	0.00
1B11 ₂ -1B12	2.09	2.46	2.05	2.37	0.10	0.09	0.13	0.00	0.13
TOTALS					12.49		30.24	15.13	15.11

FIGURE 1C

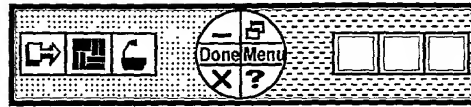


FIGURE 2A1

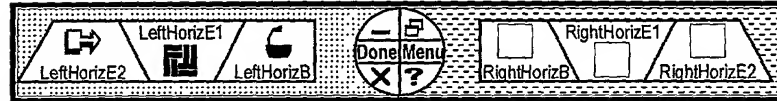


FIGURE 2A2

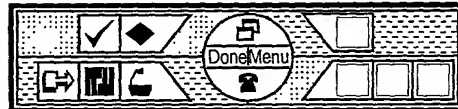


FIGURE 2B1

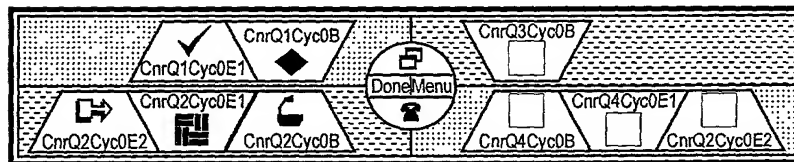


FIGURE 2B2

FIG. 2A1

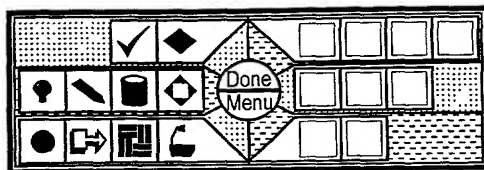


FIGURE 2C1

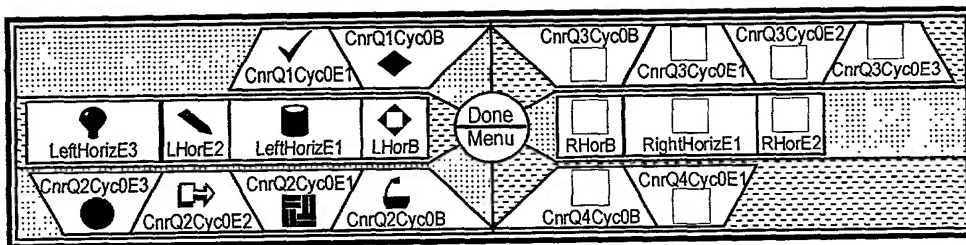


FIGURE 2C2

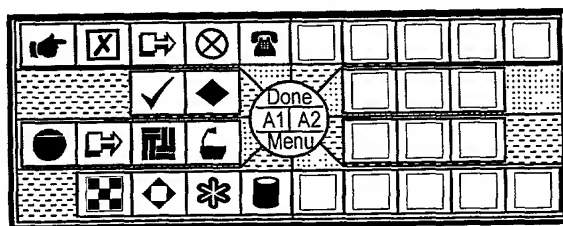


FIGURE 2D1

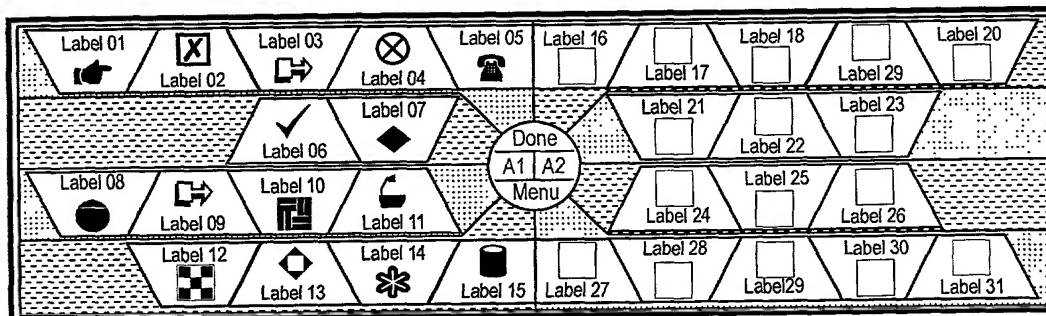


FIGURE 2D2

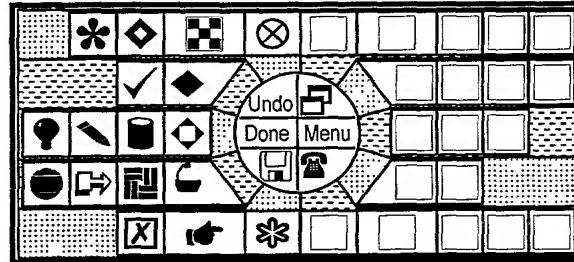


FIGURE 2E1

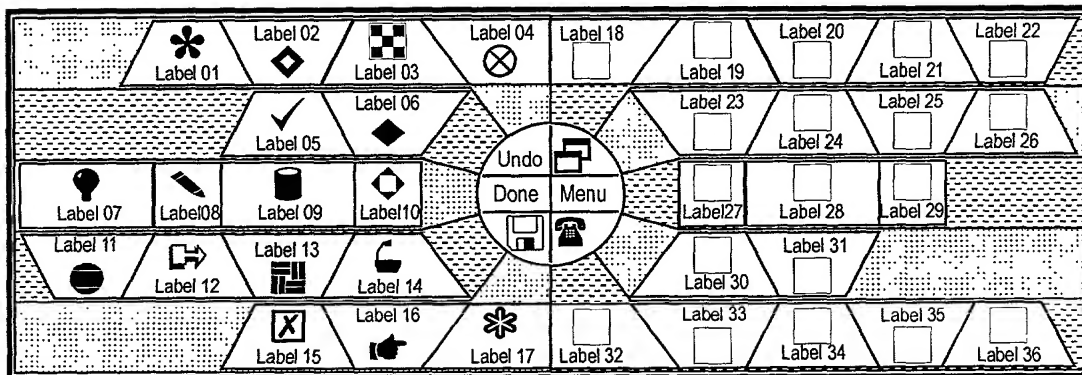


FIGURE 2E2

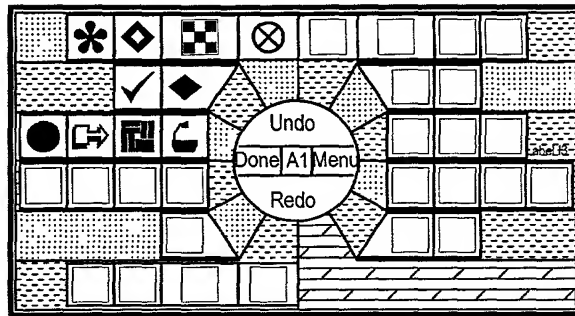


FIGURE 2F1

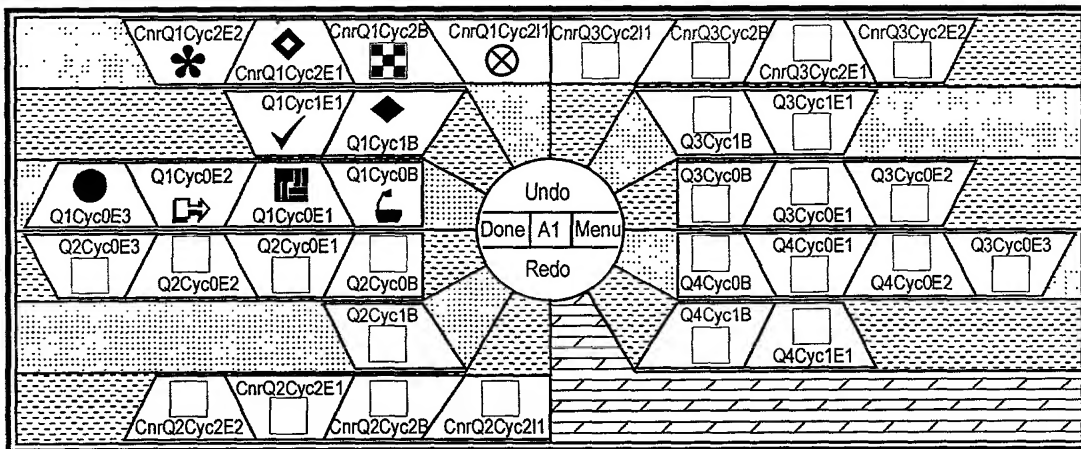


FIGURE 2F2

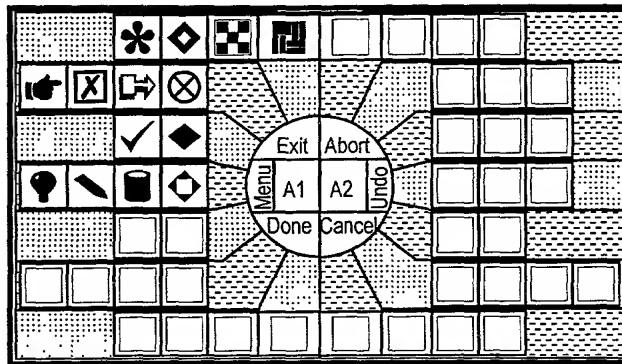


FIGURE 2G1

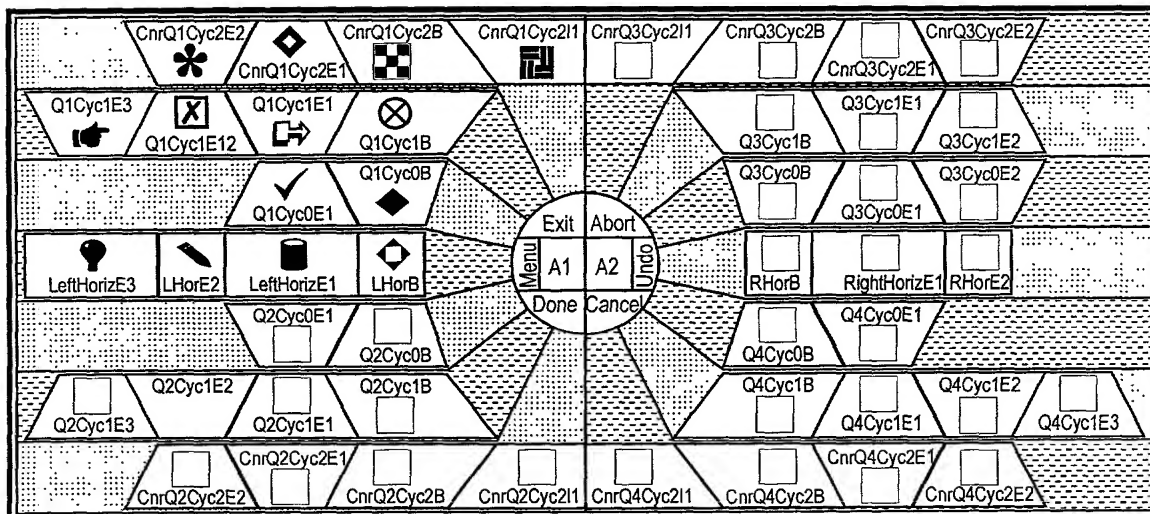


FIGURE 2G2

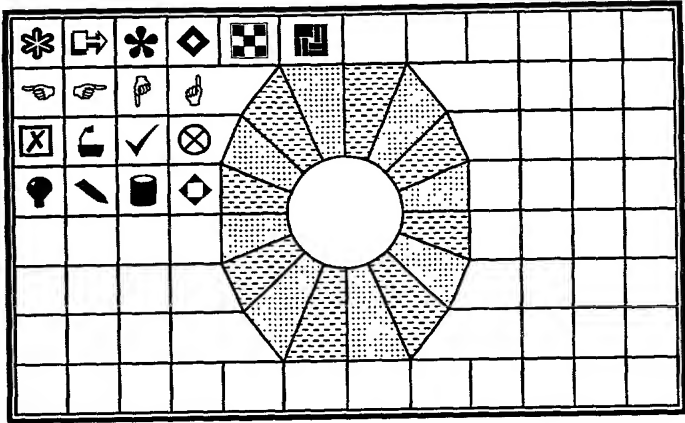


FIGURE 2H1

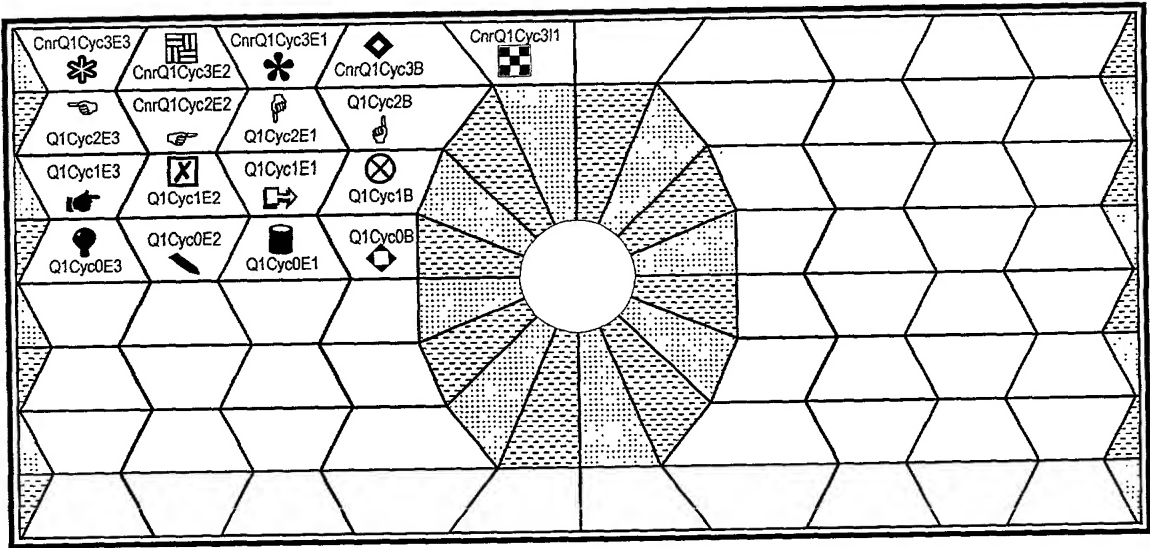


FIGURE 2H2

FIG. 3A

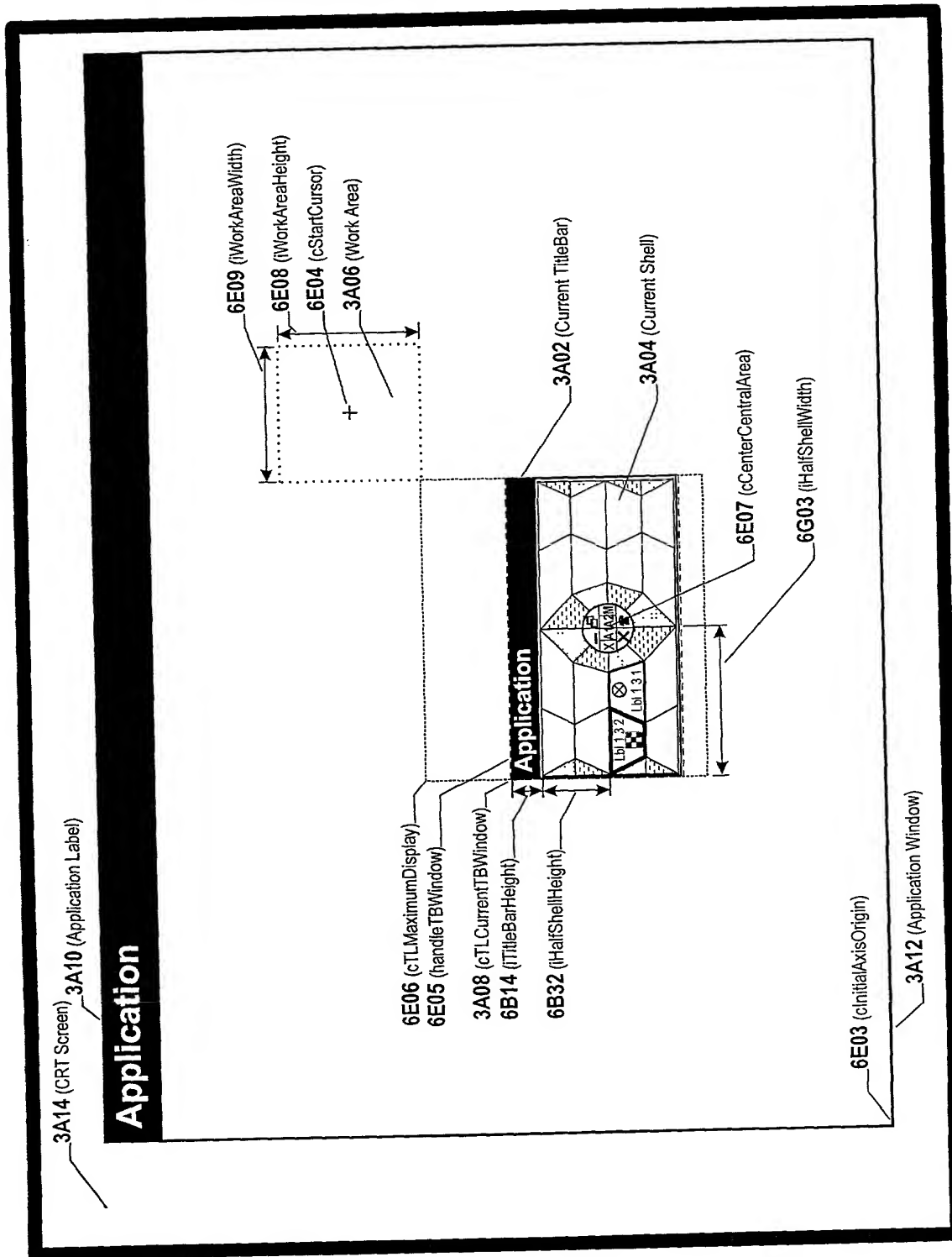


FIGURE 3A

FIG. 3B

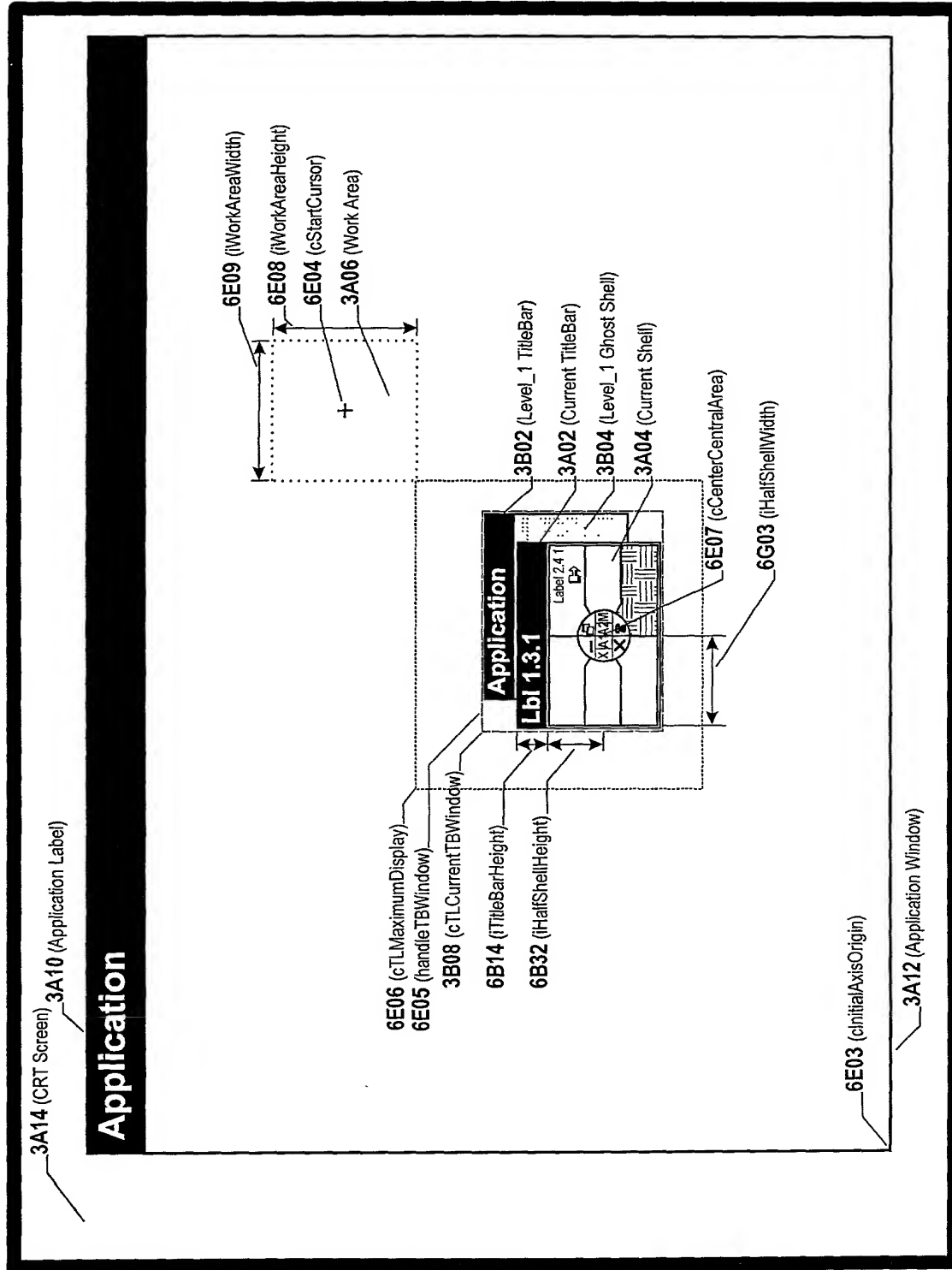


FIGURE 3B

12/68

FIGURE 3C

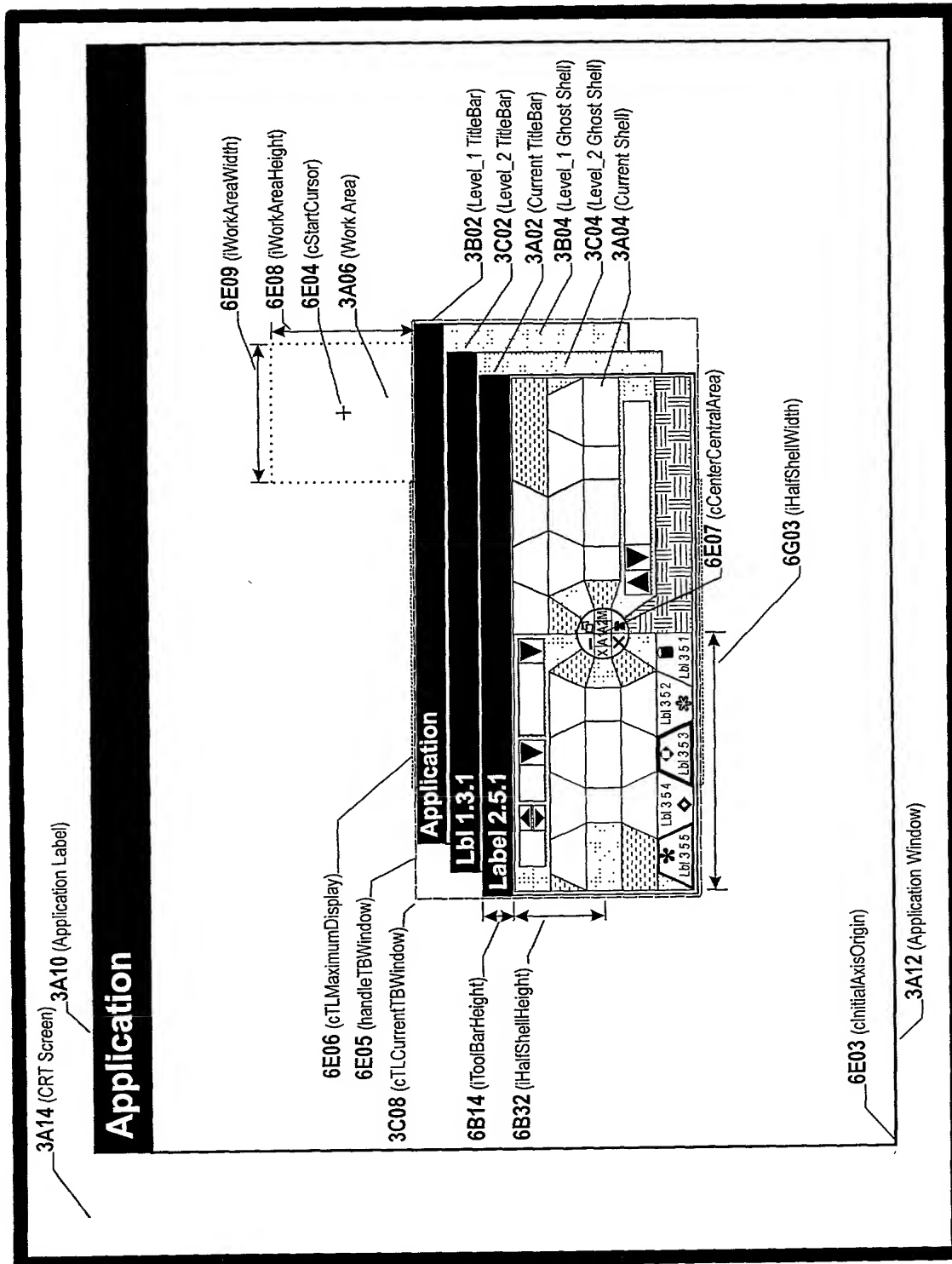


FIGURE 3C

FIG. 1

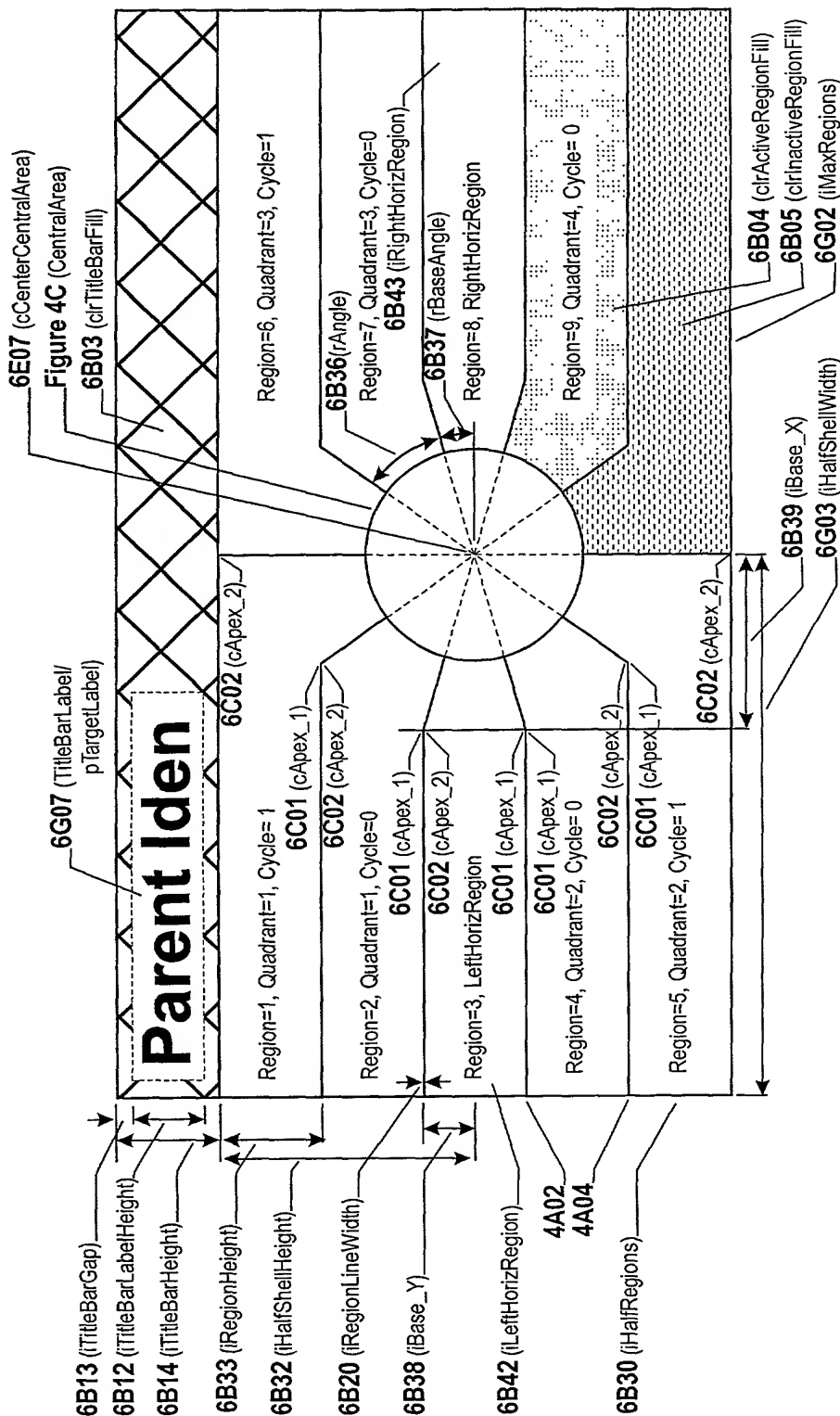


FIGURE 4A

FIGURE 4B

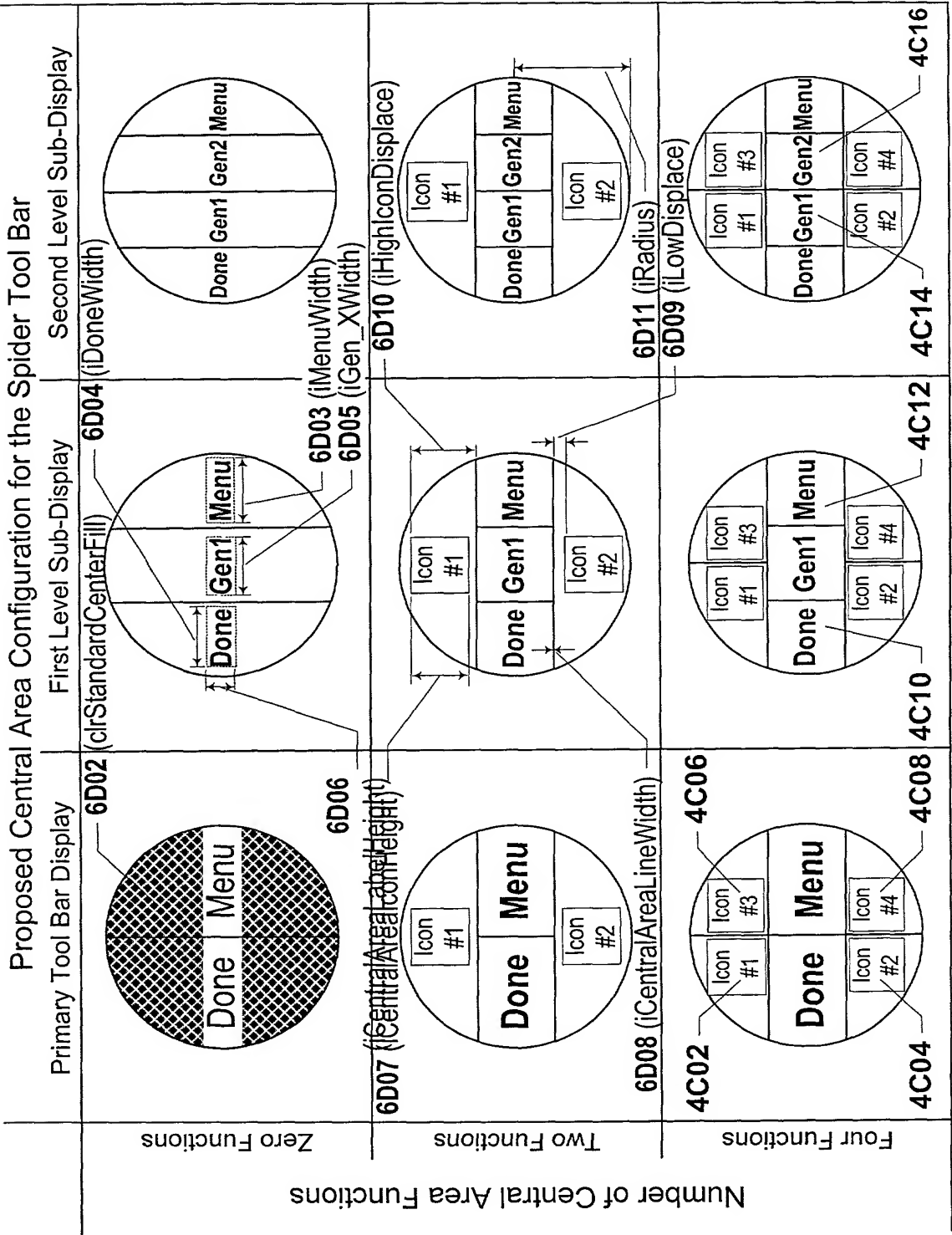


FIGURE 4C

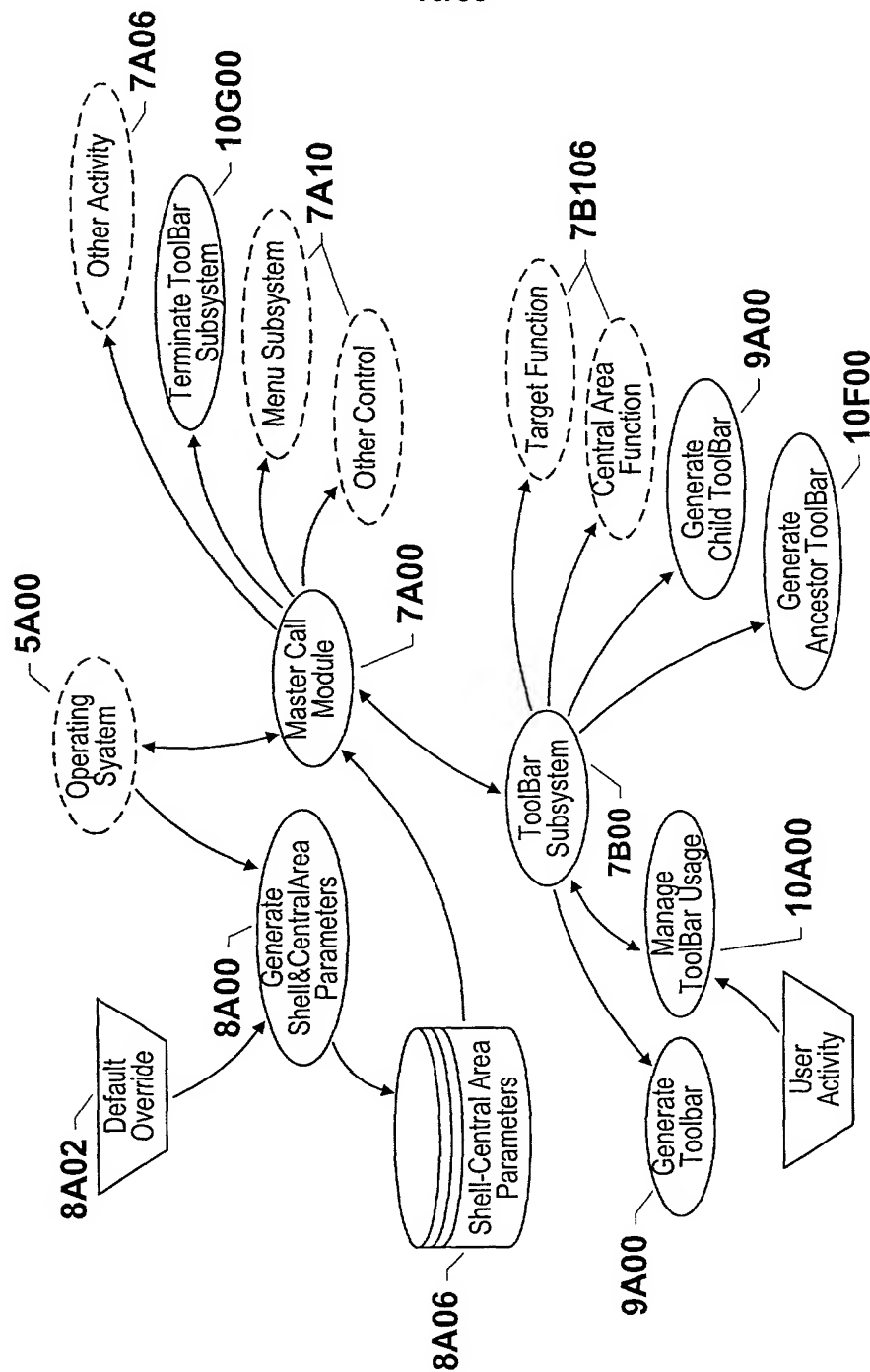


FIGURE 5A

FIGURE 5B

FIG. 5C

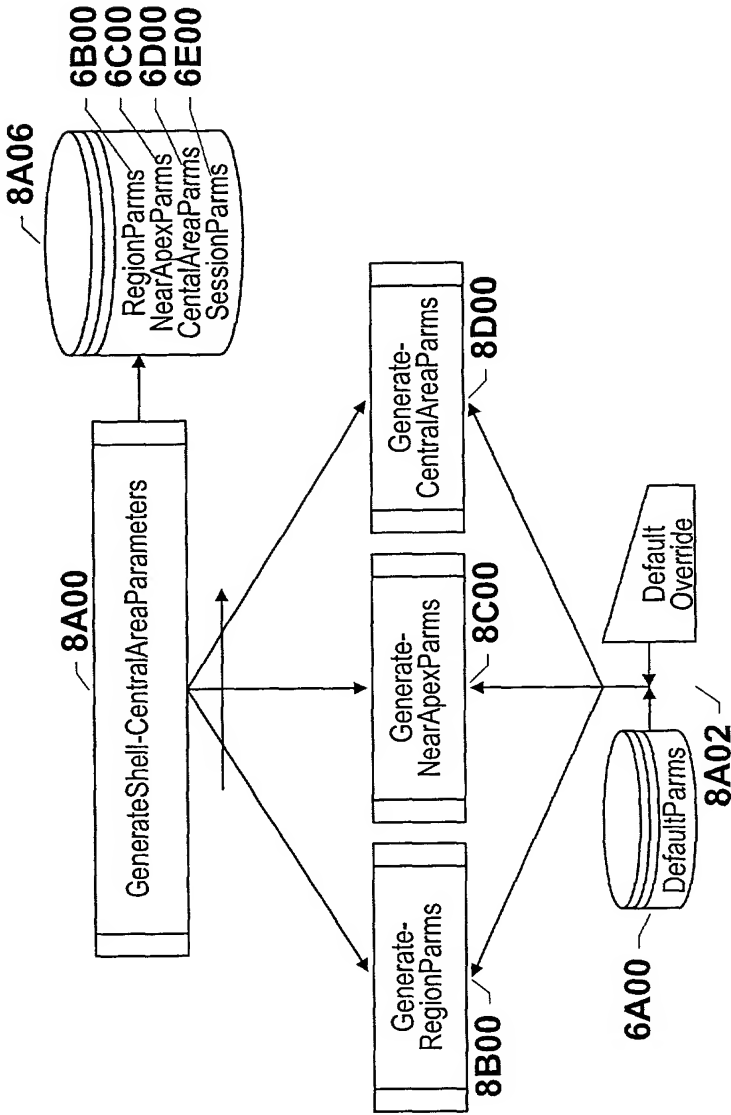


FIGURE 5C

FIG. 5D

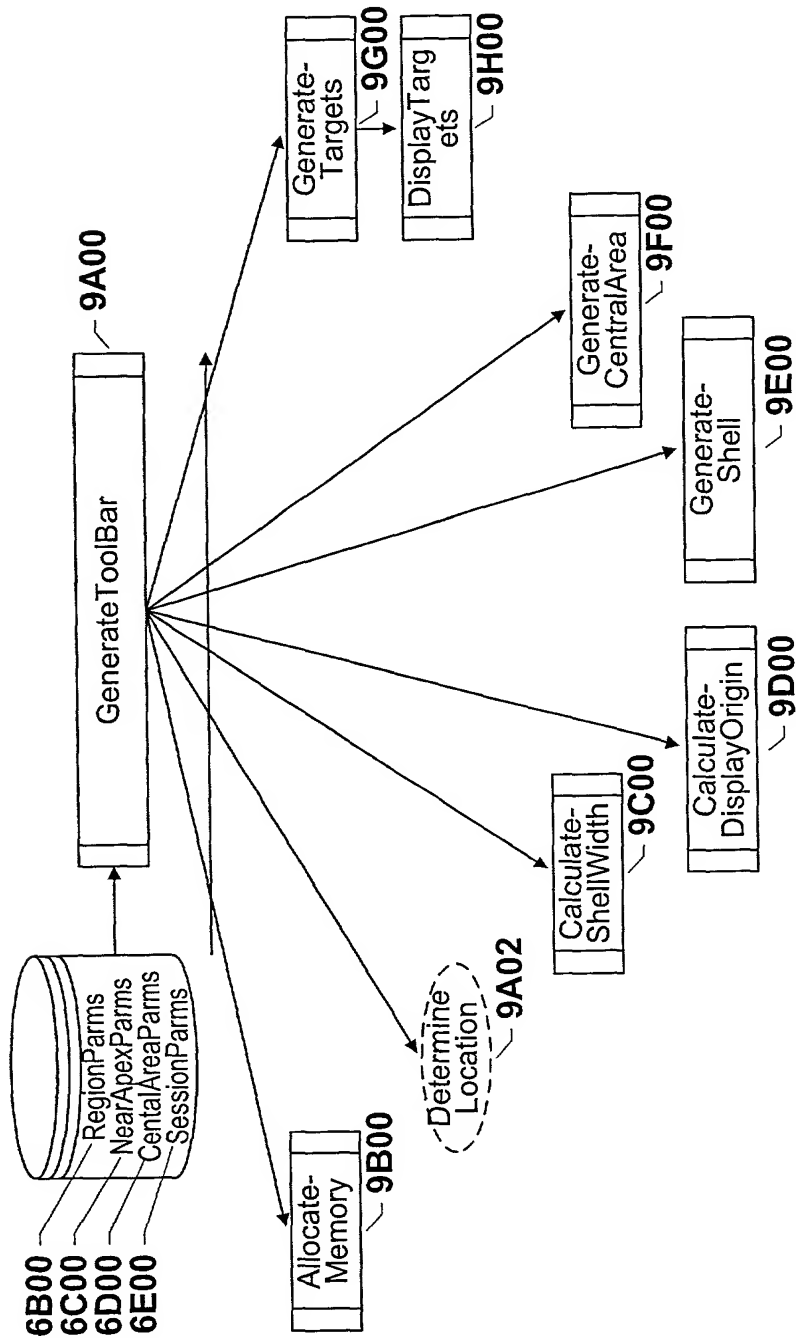
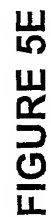


FIGURE 5D



	rDefaults	Where Stored	Preferred Default
01	iWorkAreaHeight	6E08	1 inch
02	iWorkAreaWidth	6E09	2 inches
03	sTargetShape	see 8B104	{Trapezoid Rectangle}
04	sRemoteTargetCriterion	6E11	{Most Used User Declare}
05	sPrimaryTargetIden	see 8B102	{Fill Edge Both}
06	sSecondaryTargetIden	see 8B102	{Fill Edge Both}
07	rTimeOut	6E12	0.05 second
08	iMinCursorMove	6E13	1 pixel
09	bJumpCursor	6E14	{TRUE FALSE}
10	iRegionsLimit	6B01	14 regions
11	iMaxGenerations	6B02	(application specific)
12	sTitleBarFill	6B03	Dark Blue
13	sActiveRegionFill	6B04	Light Gray
14	sInactiveRegionFill	6B05	Medium Gray
15	sPrimaryColor	see 8B102	Pale Red
16	sSecondaryColor	see 8B102	Pale Green
17	sStandardColor	6B10	Light Gray
18	sHostingFill	6B11	Dark Blue
19	iTitleBarLabelHeight	6B12	10 points
20	iTitleBarGap	6B13	2 points
21	iShellLabelHeight	6B16	10 points with trap.; 0 with rect.
22	iShellIconHeight	6B17	12 points
23	iRegionGap	6B34	1 point
24	iTargetGap	6B18	1 point
25	iIntraGap	6B19	2 points with trap.; 0 with rect.
26	iRegionLineWidth	6B20	2 points
27	iTargetLineWidth	6B21	2 points
28	rAspectRatio	6B23	2
29	iDefaultBase_X	6B40	24 points
30	sStandardCenterFill	6D02	Light Blue
31	iCentralAreaLabelHeight	6D06	10 points
32	iCentralAreaIconHeight	6D07	16 points
33	iCentralAreaLineWidth	6D08	2 points
34	iIconGap	6D09	2 points
35	iRadius	6D11	36 points
36	sCentralFunc_1	6D14 (system)	Minimize active window
37	sCentralFunc_2	6D15 (system)	Maximize active window
39	sCentralFunc_3	6D16 (system)	Close active window
40	sCentralFunc_4	6D17 (system)	Help

6A00

FIGURE 6A

rRegions

rFixedParms	aVarParms
-------------	-----------

6B00

rFixedParms

01	iRegionsLimit
02	iMaxGenerations
03	clrTitleBarFill
04	clrActiveRegionFill
05	clrInactiveRegionFill
06	clrPrimaryFill
07	clrPrimaryLine
08	clrSecondaryFill
09	clrSecondaryLine
10	clrStandardTarget
11	clrHostingFill
12	iTitleBarLabelHeight
13	iTitleBarGap
14	iTitleBarHeight
15	iMaxLabelLength
16	iShellLabelHeight
17	iShellIconHeight
18	iTargetGap
19	iIntraGap
20	iRegionLineWidth
21	iTargetLineWidth
22	iTargetHeight
23	iAspectRatio
24	iHalfNarrowSide
25	iHalfWideSide
26	iInterTargetDistance
27	iLowLabelDisplace_Y
28	iLowIconDisplace_Y
29	iHighDisplace_Y

aVarParms_R

empty ₁
rVariables ₂
empty ₃
rVariables ₄
empty ₅
:
rVariables _{iRegionsLimit}

rVariables_K

30	iHalfRegions
31	iNumCycles
32	iHalfShellHeight
33	iRegionHeight
34	iRegionGap
35	iTargetDisplace_Y
36	rAngle
37	rBaseAngle
38	iBase_Y
39	iBase_X
40	iDefaultBase_X
41	iMidLine
42	iLeftHorizRegion
43	iRightHorizRegion
44	aQRegion[1]
45	aQRegion[2]
46	aQRegion[3]
47	aQRegion[4]

FIGURE 6B1

DEFINITIONS

rRegionParms.FixedParms: Parameters applicable to all shells

- | | | |
|-----------|------------------------------|--|
| 01 | iRegionsLimit | Maximum number of regions permitted for any display. |
| 02 | iMaxGenerations | Maximum number of generations permitted per toolbar activation. |
| 03 | clrTitleBarFill | Background color of the title bar. |
| 04 | clrActiveRegionFill | Background color for regions that contain targets. |
| 05 | clrInActiveRegionFill | Background color for regions that do not contain targets. |
| 06 | clrPrimaryFill | The color designating the color of the primary target fill. |
| 07 | clrPrimaryLine | The color designating the color of the primary target boundary. |
| 08 | clrSecondaryFill | The color designating the color of the secondary target fill. |
| 09 | clrSecondaryLine | The color designating the color of the secondary target boundary. |
| 10 | clrStandardTarget | The color designating the color of all standard targets. |
| 11 | clrHostingFill | Background color for display shape currently hosting cursor. |
| 12 | iTitleBarLabelHeight | Height of the label in the titlebar. |
| 13 | iTitleBarGap | Space between titlebar boundary and the bounding rectangle of the titlebar label. |
| 14 | iTitleBarHeight | Height of the titlebar |
| 15 | iMaxLabelLength | If target labels are present, the maximum permitted length of a target label. |
| 16 | iShellLabelHeight | If present, the height of text identifying target. |
| 17 | iShellIconHeight | Height and width of target icons; icons presumed to be square. |
| 18 | iTargetGap | Distance between the top of the target line and the bottom of the closest target identifier |
| 19 | ilntraGap | If present, vertical distance between top edge of bottom identifier and bottom edge of top identifier. |
| 20 | iRegionLineWidth | Thickness of line delimiting regions. |
| 21 | iTargetLineWidth | Thickness of line delimiting targets. |
| 22 | Target Height | Vertical distance between wires denoting top and bottom of target. |
| 23 | iAspectRatio | Ratio between the length of the iHalfWideSide and iHalfNarrowSide. |
| 24 | iHalfNarrowSide | One half the length of the bottom edge of the reference base target. |
| 25 | iHalfWideSide | One half the length of the top edge of the reference base target. |
| 26 | iInterTargetDistance | The horizontal distance between the midpoints of contiguous exterior targets. |
| 27 | iLowLabelDisplace_Y | Vertical distance between the bottom target wire and top of label when label is bottom identifier. |
| 28 | iLowIconDisplace_Y | Vertical distance between the bottom target wire and top of icon when icon is bottom identifier. |
| 29 | iHighDisplace_Y | Vertical distance between the bottom target wire and top of top identifier. |

FIGURE 6B2

FIGURE 6B2

rRegionParms.aVarParms[K].rVariables: Parameters for shells with K regions

- 30** iHalfRegionsNumber of regions comprising each half of the shell
- 31** iNumCycles Number of regions in a quadrant. (Horizontal regions are excluded).
- 32** iHalfShellHeight Vertical distance between the central-area origin and the top of the shell
- 33** iRegionHeight Vertical distance between the horizontal region wires in the non-converging portion of a region.
- 34** iRegionGap Vertical distance between top of region line and bottom of the target line
- 35** iTargetDisplace_Y Vertical distance between the bottom region wire and the bottom target wire.
- 36** rAngle Angle subtended by each region at center of display.
- 37** rBaseAngle Reference angle:
= 0 if no horizontal regions,
= rAngle/2 if horizontal regions are present.
- 38** iBase_Y Reference vertical displacement;
= 0 when horizontal regions are not present,
= iHalfRegionHeight when horizontal regions are present.
- 39** iBase_X Reference horizontal displacement;
iNumRegions > 4 regions: = Apex2.X for region containing rBaseAngle > 0.
iNumRegions <= 4 regions:= rRadius + iDefaultBase_X.
- 40** iDefaultBase_X Externally assigned displacement from edge of central area. Employed with 2 & 4 region displays.
- 41** iMidLine The datum line for determination of the horizontal center of exterior targets
- 42** iLeftHorizRegion If present, region number of the left horizontal region.
- 43** iRightHorizRegion If present, region number of the right horizontal region.
- 44** aQRegion[1] Number regions to left of the Y-axis and wholly above the X-axis
- 45** aQRegion[2] Number regions to left of the Y-axis and wholly below the X-axis
- 46** aQRegion[3] Number regions to right of the Y-axis and wholly above the X-axis
- 47** aQRegion[4] Number regions to right of the Y-axis and wholly below the X-axis

FIGURE 6B3

aApexParms 6C00

empty ₁
aRegionApexes ₂
empty ₃
aRegionApexes ₄
empty ₅
:
aRegionApexes _{iRegionsLimit}

aRegionApexes_R

rNearApexes ₁
rNearApexes ₂
:
rNearApexes _{iMaxRegions}
null _{iMaxRegions+1}
:
null _{iRegionsLimit}

rNearApexes_C

01	cApex_1
02	cApex_2
03	iNumInteriorTargets

cApex_N

X	Y
---	---

DEFINITIONS

aApexParms[R].aRegionApexes[C].rNearApexes

- 01 cApex_1 The bottom apex closest to X-axis of the Cth region of a R region shell.
02 cApex_2 The top apex closest to X-axis of the Cth region of a R region shell.
03 iNumInteriorTargets Number of targets that can be displayed between iBase_X and the Y-axis.

FIGURE 6C

FIGURE 6C

26/68

rCentralArea

rFixedParms	aFuncs
-------------	--------

6D00

rFixedParms

01	iNumFuncs
02	clrStandardCenterFill
03	iHalfMenuWidth
04	iHalfDoneWidth
05	iHalfGen_XWidth
06	iCentralAreaLabelHeight
07	iCentralAreaIconHeight
08	iCentralAreaLineWidth
09	iIconGap
10	iHighIconDisplace
11	iRadius

aFuncs

12	rLocator _{Menu}
13	rLocator _{Exit}
14	rLocator _{Func_1}
15	rLocator _{Func_2}
16	rLocator _{Func_3}
17	rLocator _{Func_4}

rLocator

18	pFunc
19	plcon

DEFINITIONS

rCentralArea.rFixedParms

01	iNumFuncs	Number of function icons defined for the central-area.
02	clrStandardCenterFill	Background color for central-area shapes that are not hosting the cursor.
03	iHalfMenuWidth	Half length of the word "Menu"
04	iHalfDoneWidth	Half length of the word "Done"
05	iHalfGen_XWidth	Half length of the string "Gen_X"
06	iCentralAreaLabelHeight	Height of text of the central-area.
07	iCentralAreaIconHeight	Height of icon in the central-area.
08	iCentralAreaLineWidth	Thickness of line delimiting shapes within the central-area.
09	iIconGap	Distance from the relevant central area line and the near edge of the icon.
10	iHighIconDisplace	Distance from the relevant line wire to most distant icon edge icon.
11	iRadius	Radius of arc portions of the central-area.

rCentralArea.rFuncs

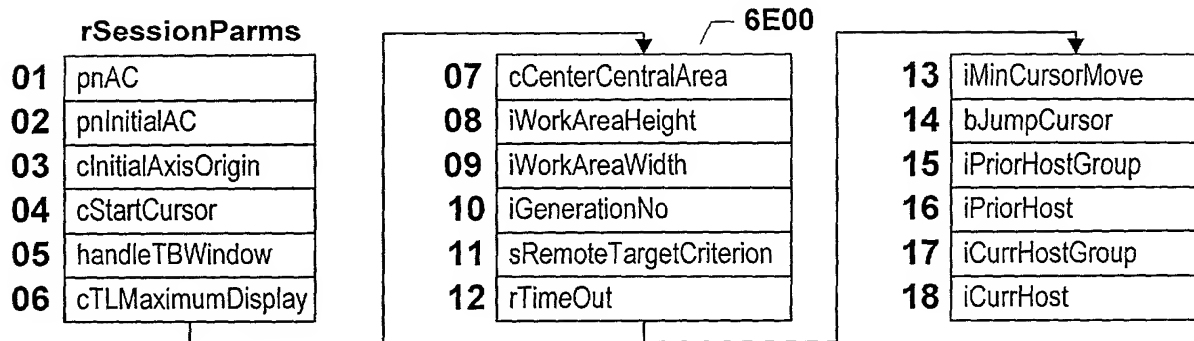
12	rLocator _{Menu}	Pointer to the Menu subsystem. There is no icon pointer
13	rLocator _{Exit}	Terminates the current activation of the toolbar subsystem via 10G00. This record contains null values and is provided only to maintain logical symmetry.
14	rLocator _{Func_1}	If present: pFunc-> Address Function #1; plcon-> Address Icon #1; null otherwise.
15	rLocator _{Func_2}	If present: pFunc-> Address Function #2; plcon-> Address Icon #2; null otherwise.
16	rLocator _{Func_3}	If present: pFunc-> Address Function #3; plcon-> Address Icon #3; null otherwise.
17	rLocator _{Func_4}	If present: pFunc-> Address Function #4 plcon-> Address Icon #4; null otherwise.

rCentralArea.rFuncs[N].rLocator

18	pFunc	Pointer to module implementing the N th central-area function.
19	plcon	Pointer to icon of N th central-area function.

FIGURE 6D

FIGURE 6D

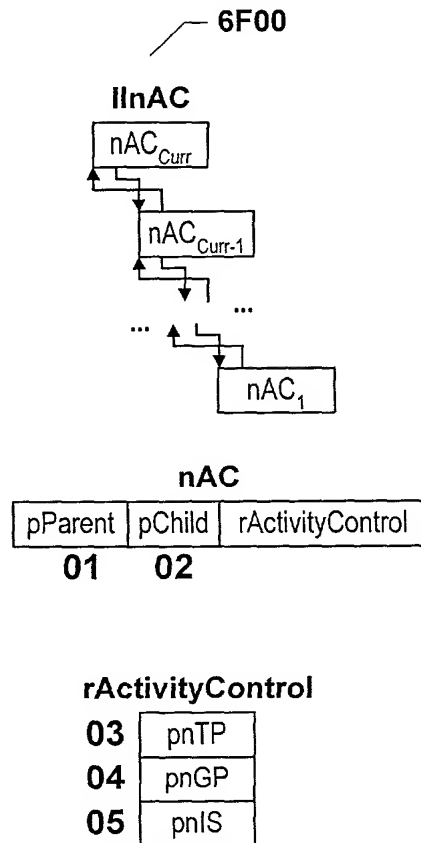


DEFINITIONS

rSessionParms: ARRAY OF VARIABLES VALID FOR A CURRENTLY ACTIVE DISPLAY

01 pnAC	Pointer to the llnAC array of pointers that identify nodes of linked list structures that control user interaction with the current activation of the toolbar subsystem.
02 pnInitialAC	Pointer to first node of llnAC. Allocated at each activation of toolbar subsystem.
03 clnitialAxisOrigin	Screen coordinates of axis at activation if toolbar subsystem.
04 cStartCursor	Cursor coordinates at toolbar subsystem activation (relative to clnitialAxisOrigin).
05 handleTBWindow	Pointer to data and procedures that manage the display window.
06 cTLMaximumDisplay	Coordinates of top-left corner of maximum display (relative to cCenterCentralArea).
07 cCenterCentralArea	Coordinates of the center of the central-area (relative to clnitialAxisOrigin).
08 iWorkAreaHeight	Height of screen area of current user interest.
09 iWorkAreaWidth	Width of screen area of current user interest.
10 iGenerationNo	Number of generations of current toolbar display.
11 sRemoteTargetCriterion	String identifying criterion for primary and secondary targets of each group.
12 rTimeOut	Duration between "time-outs" used to check for cursor movement
13 iMinCursorMove	The least distance the cursor can move without testing for new host shape.
14 bJumpCursor	TRUE= jump cursor to cStartCursor at termination of toolbar subsystem, FALSE= do not jump cursor at termination of toolbar subsystem.
15 iPriorHostGroup	Identifier of region hosting cursor prior to current host (null if a non-Region).
16 iPriorHost	Identifier of cursor host that preceeded current host .
17 iCurrHostGroup	Identifier of region currently hosting cursor (null if a non-Region).
18 iCurrHost	Identifier of current cursor host:
	>100 (Icon lden+100) Identification of target
	(iNumRegions+1) to >iMaxRegions Identification of inactive region
	1 to iNumRegions Identification of active region
	0 Title bar of current toolbar
	-1 & -2 Gen_1 & Gen_2 shape respectively
	-3 Menu shape
	-4 Done shape
	-5 to -8 #1-#4 central-area function shapes
	-11 to -12 -(Generation+10) Label of ancestors 1 & 2 respectively
	-21 to -22 -(Generation+20) Ghost shells of ancestors 1 & 2 respectively

FIGURE 6E



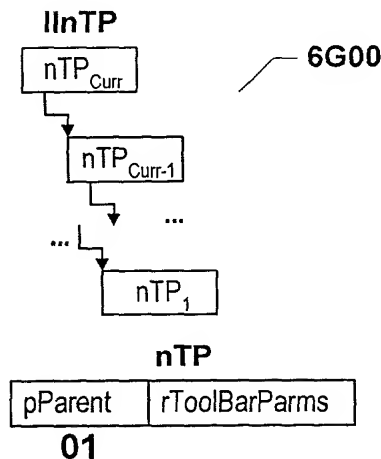
DEFINITIONS

- 01 pParent Pointer to parent of current nAC node
 02 pChild Pointer to child of current nAC node.

***pnAC.rActivityControl.**

- 03 pnTP Pointer to currently active nTP node.
 04 pnGP Pointer to currently active nGP node.
 05 pnIS Pointer to currently active nIS node.

FIGURE 6F



rToolBarParms	
02	iNumRegions
03	iHalfShellWidth
04	iNumTargetGroups
05	iTotalNumTargets
06	iSelectedTargetIden
07	sToolBarLabel
08	cTLShell
09	cTLDisplay
10	bDoneButton

DEFINITIONS

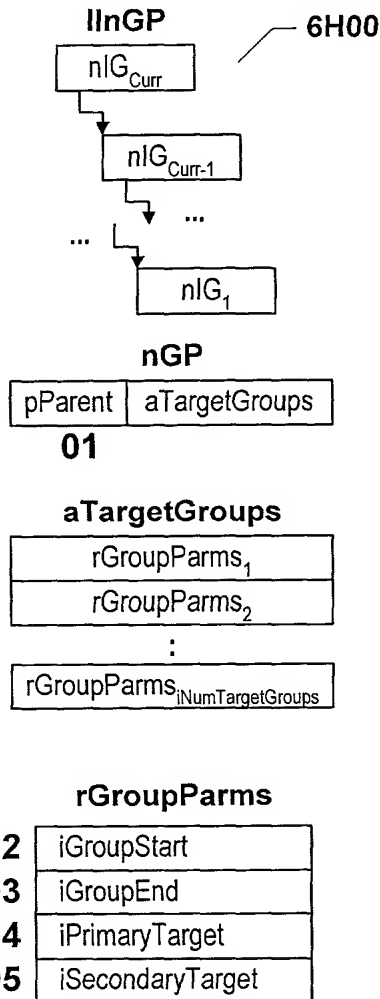
01 pParent Pointer to parent of current nTP node.

*pnTP.rToolBarParms

02 iNumRegions Number of regions of the current display.
03 iHalfShellWidth Half the width of the currently active shell.
04 iNumTargetGroups Number of target groups in current toolbar.
05 iTotalNumTargets Total number of targets in current toolbar.
06 iSelectedTargetIden Identification number of target selected by the user.
07 sToolBarLabel Label identifying the current toolbar; null if no label.
08 cTLShell Coordinates of current shell origin
09 cTLDisplay Coordinates of display origin
10 bDoneButton T-> "Done" shape in central-area;
 F-> no "Done" shape in central-area.

FIGURE 6G

30/68



DEFINITIONS

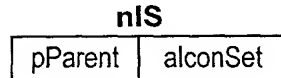
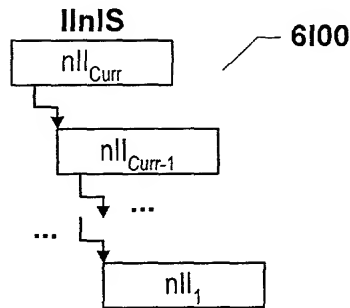
01 pParent Pointer to parent of current nGP node.

*pnGP.aTargetGroups[K].rGroupParms

- 02 iGroupStart Target identifier number of current nIS.alconSet commencing the Kth group.
- 03 iGroupEnd Target identifier number of current nIS.alconSet ending the Kth group.
- 04 iPrimaryTarget Target identifier number of primary target of Kth target group.
- 05 iSecondaryTarget Target identifier number of secondary target of Kth target group.

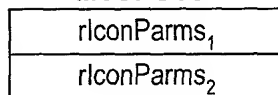
FIGURE 6H

31/68

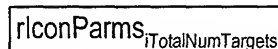


01

alconSet



:



rlconParms

02	pChildToolbar
03	pOtherControl
04	iMenuTag
05	pIcon
06	pLabel
07	iTargetType

DEFINITIONS

01 pParent Pointer to parent of current nIS node.

***pnlS.alconSet[K].rlconParms**

- 02 pChildToolbar Pointer to icon subset when selected target activates another level of display
Null if no child toolbar.
- 03 pOtherControl Pointer to icon subset when selected target activates display of a non-toolbar control.
Null if no non-toolbar control.
- 04 iMenuTag Tag value of menu leaf when selected target is menu leaf. **Null** if not a menu leaf.
- 05 pIcon Pointer to icon identifier of target.
- 06 pLabel Pointer to label identifier of target.
- 07 iTargetType Identifies technique by which user acquires target:
- 1= PrimaryTarget: Acquire by remote single click or by single click on target.
 - 2= SecondaryTarget: Acquire by remote double click or by single click on target.
 - 3= StandardTarget: Acquire only by single click on target.

FIGURE 61

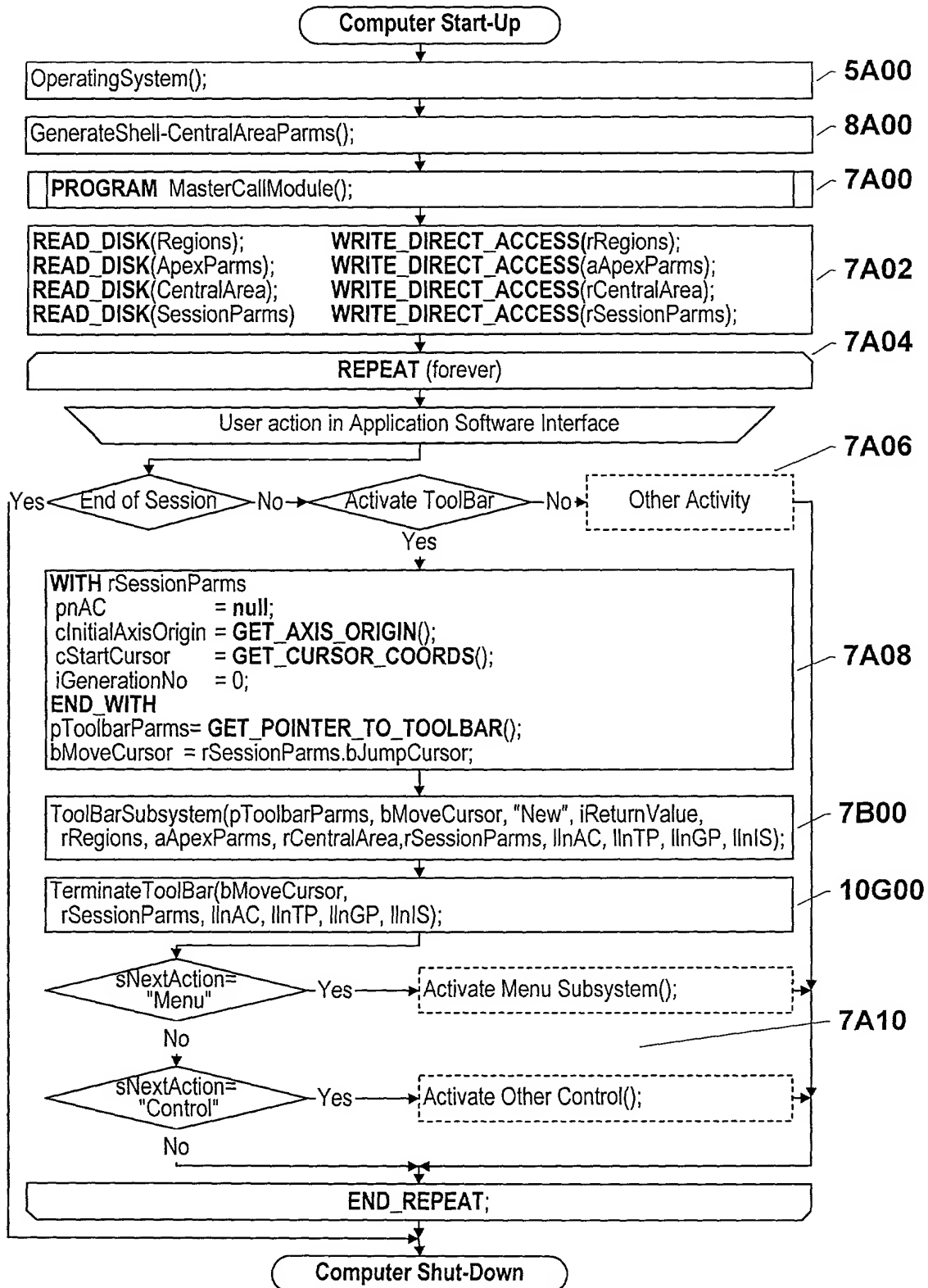


FIGURE 7A

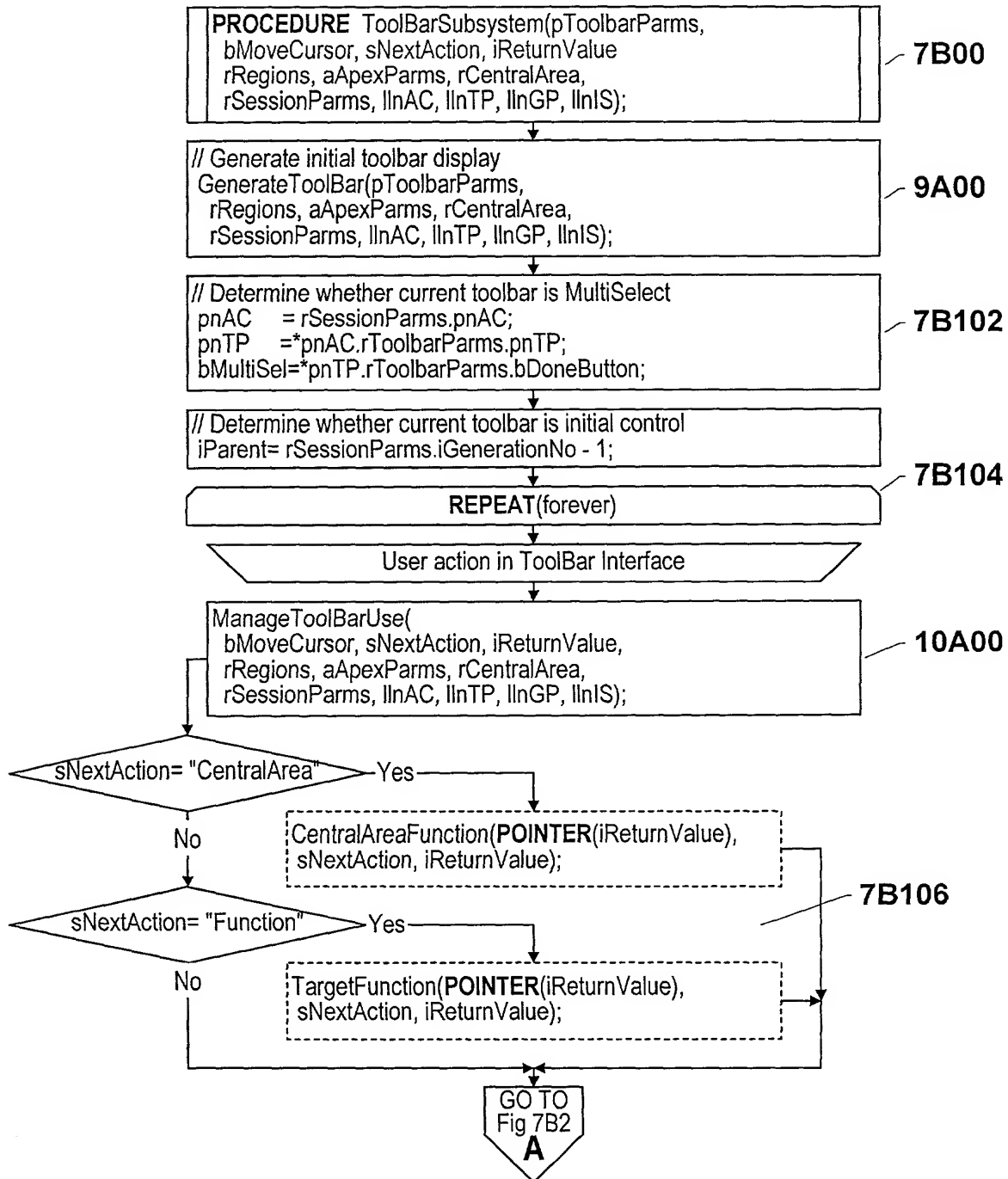


FIGURE 7B1

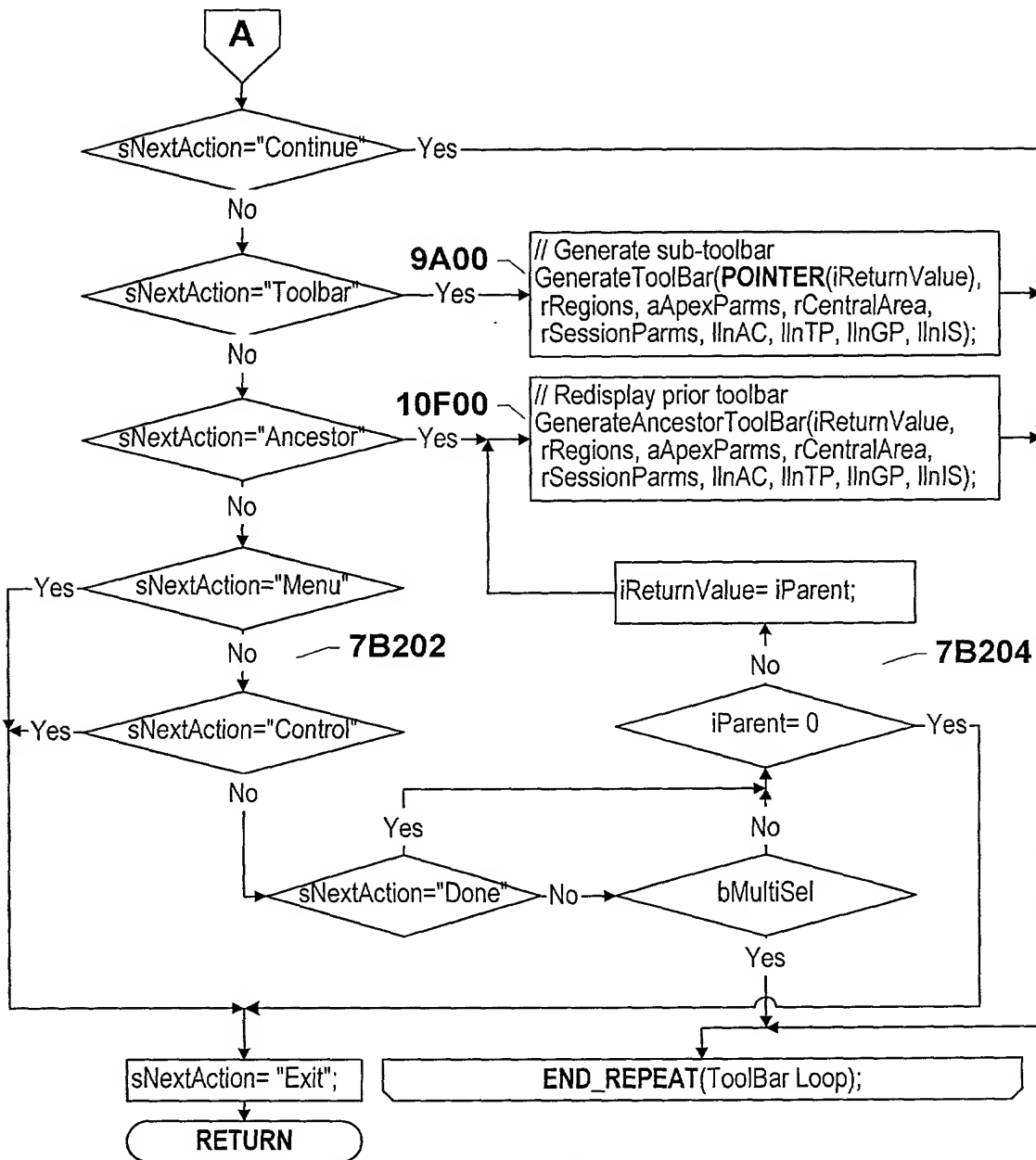


FIGURE 7B2

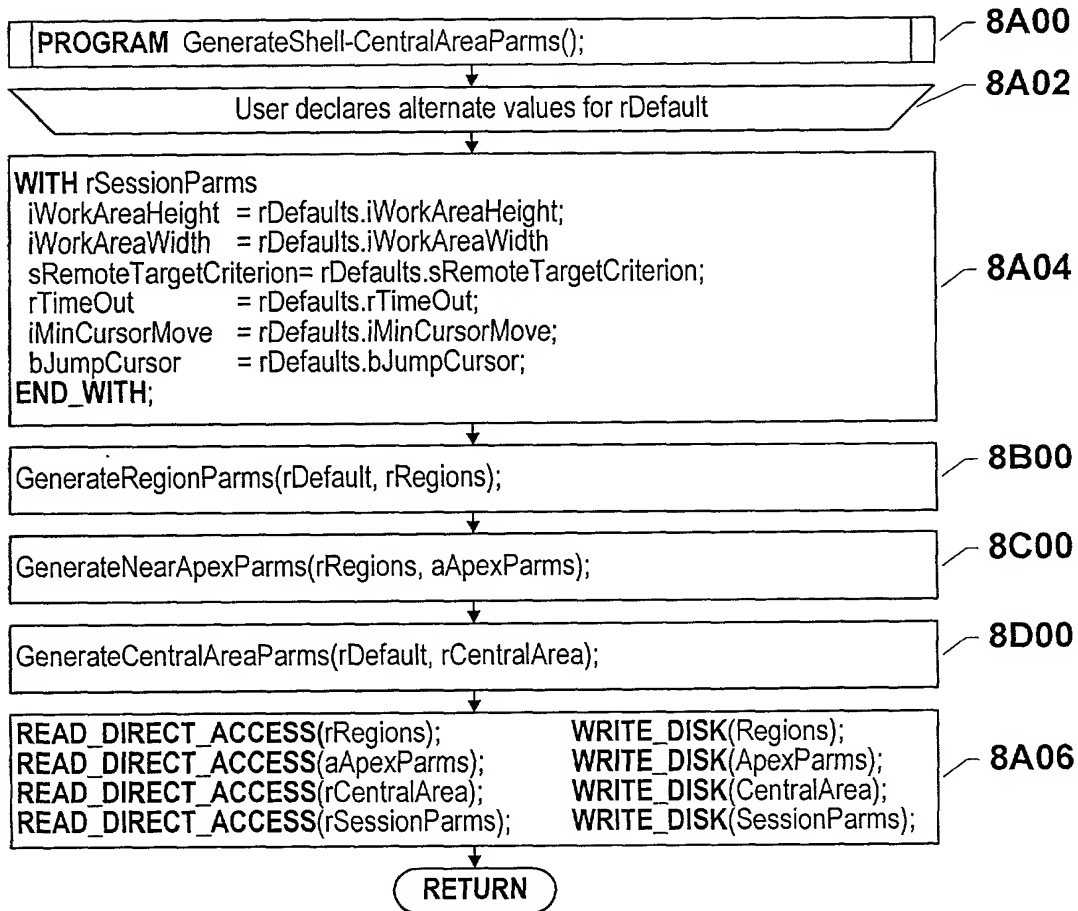


FIGURE 8A

PROCEDURE GenerateRegionParms(rDefaults, rRegions);

8B00

8B102

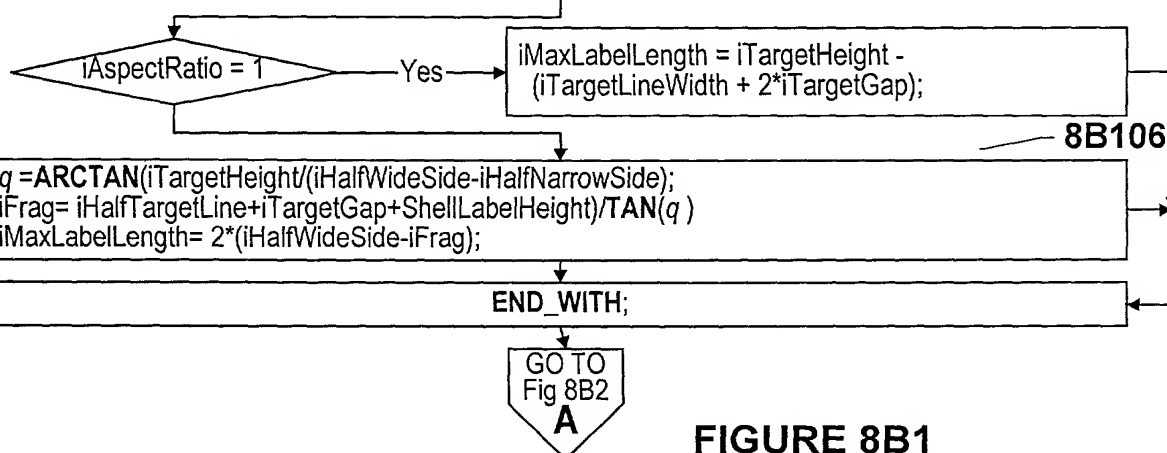
```

WITH rRegions.rFixedParms
iRegionsLimit      = rDefaults.iRegionsLimit;      iMaxGenerations= rDefaults.iMaxGenerations;
clrTitleBarFill    = COLOR(rDefaults.sTitleBarFill);
clrActiveRegionFill = COLOR(rDefaults.sActiveRegionFill);
clrInactiveRegionFill= COLOR(rDefaults.sInactiveRegionFill);
CASE(rDefaults.sPrimaryTargetIden)
  Fill: clrPrimaryFill=COLOR(rDefaults.sPrimaryColor);
  clrPrimaryLine=COLOR(rDefaults.sStandardColor);
  Edge:clrPrimaryFill=COLOR(rDefaults.sStandardColor); clrPrimaryLine=COLOR(rDefaults.sPrimaryColor);
  Both: clrPrimaryFill=COLOR(rDefaults.sPrimaryColor); clrPrimaryLine= COLOR(rDefaults.sPrimaryColor);
END CASE;
CASE(rDefaults.sSecondaryTargetIden)
  Fill: clrSecondaryFill = COLOR(rDefaults.sSecondaryColor);
  clrSecondaryLine = COLOR(rDefaults.sStandardColor);
  Edge: clrSecondaryFill = COLOR(rDefaults.sStandardColor);
  clrSecondaryLine = COLOR(rDefaults.sSecondaryColor);
  Both: clrSecondaryFill = COLOR(rDefaults.sSecondaryColor);
  clrSecondaryLine = COLOR(rDefaults.sSecondaryColor);
END CASE;
clrStandardTarget= COLOR(rDefaults.sStandardColor);
clrHostingFill    = COLOR(rDefaults.sHostingFill);
iTitleBarLabelHeight= rDefaults.iTitleBarLabelHeight;    iTitleBarGap= rDefaults.iTitleBarGap;
iShellLabelHeight= rDefaults.iShellLabelHeight;          iShellIconHeight = rDefaults.iShellIconHeight;
iTargetGap        = rDefaults.iTargetGap;                iIntraGap= rDefaults.iIntraGap;
iRegionLineWidth= rDefaults.RegionLineWidth;            iTargetLineWidth = rDefaults.iTargetLineWidth;
iLowLabelDisplace_Y= iTargetLineWidth/2 + iTargetGap +iShellLabelHeight;
iLowIconDisplace_Y = iTargetLineWidth/2 + iTargetGap +iShellIconHeight;
iHighDisplace     = iLowIconDisplace_Y + iIntraGap + iShellLabelHeight;
  
```

8B104

```

iTitleBarHeight=iTitleBarLabelHeight + 2*iTitleBarGap + iRegionLineWidth;
iTargetHeight = iTargetLineWidth +2*iTargetGap+iIntraGap+iShellLabelHeight+iShellIconHeight;
iHalfNarrowSide= (iShellIconHeight/2)+3; // "3" a pixel value
iHalfWideSide = (rAspectRatio*iTargetHeight)/2;
iInterTargetDistance= iHalfNarrowSide + iHalfWideSide;
IF(sTargetShape = "Rectangle") rAspectRatio = 1;
ELSE rAspectRatio = iHalfWideSide/iHalfNarrowSide;
END;
  
```



8B106

FIGURE 8B1

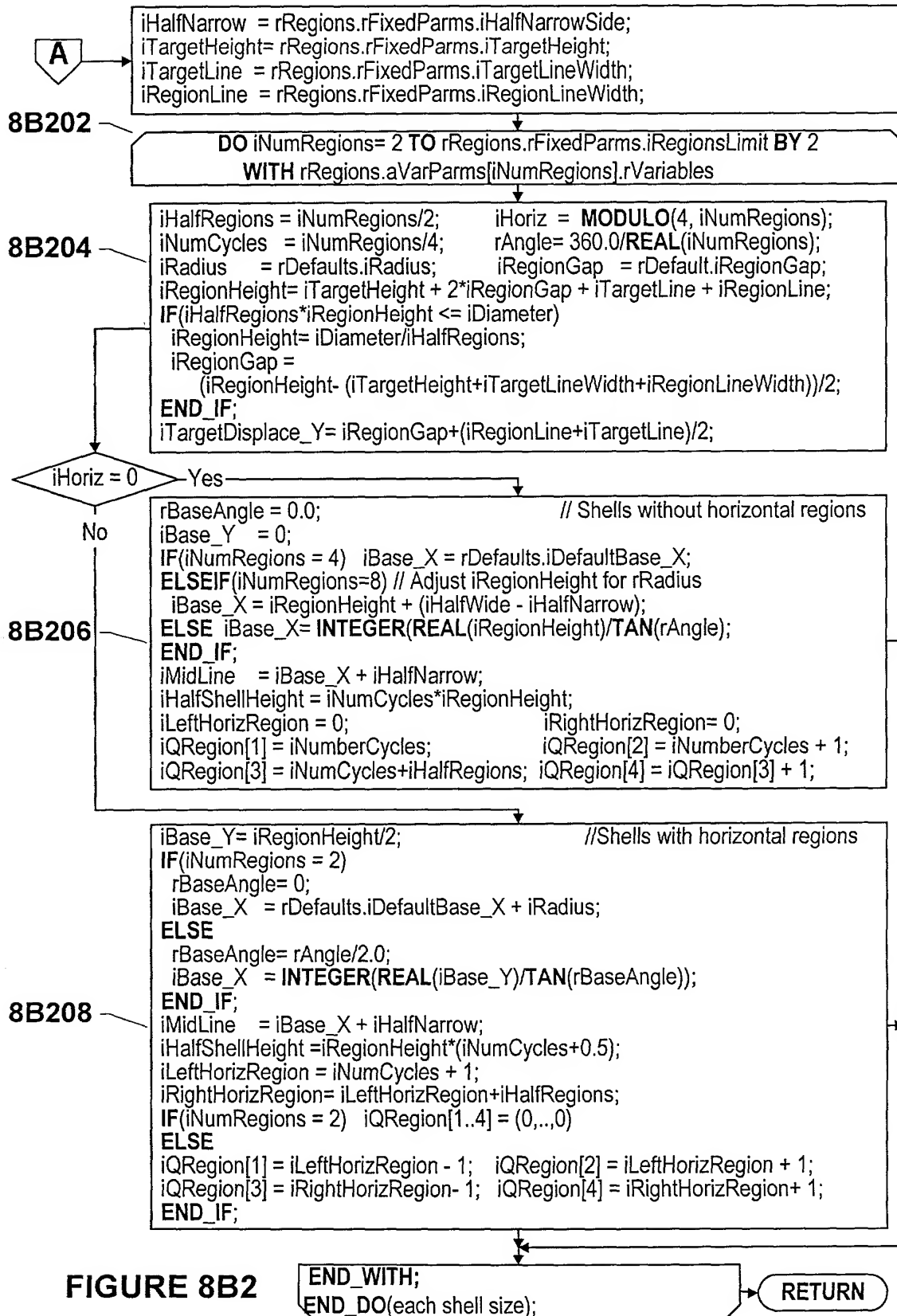
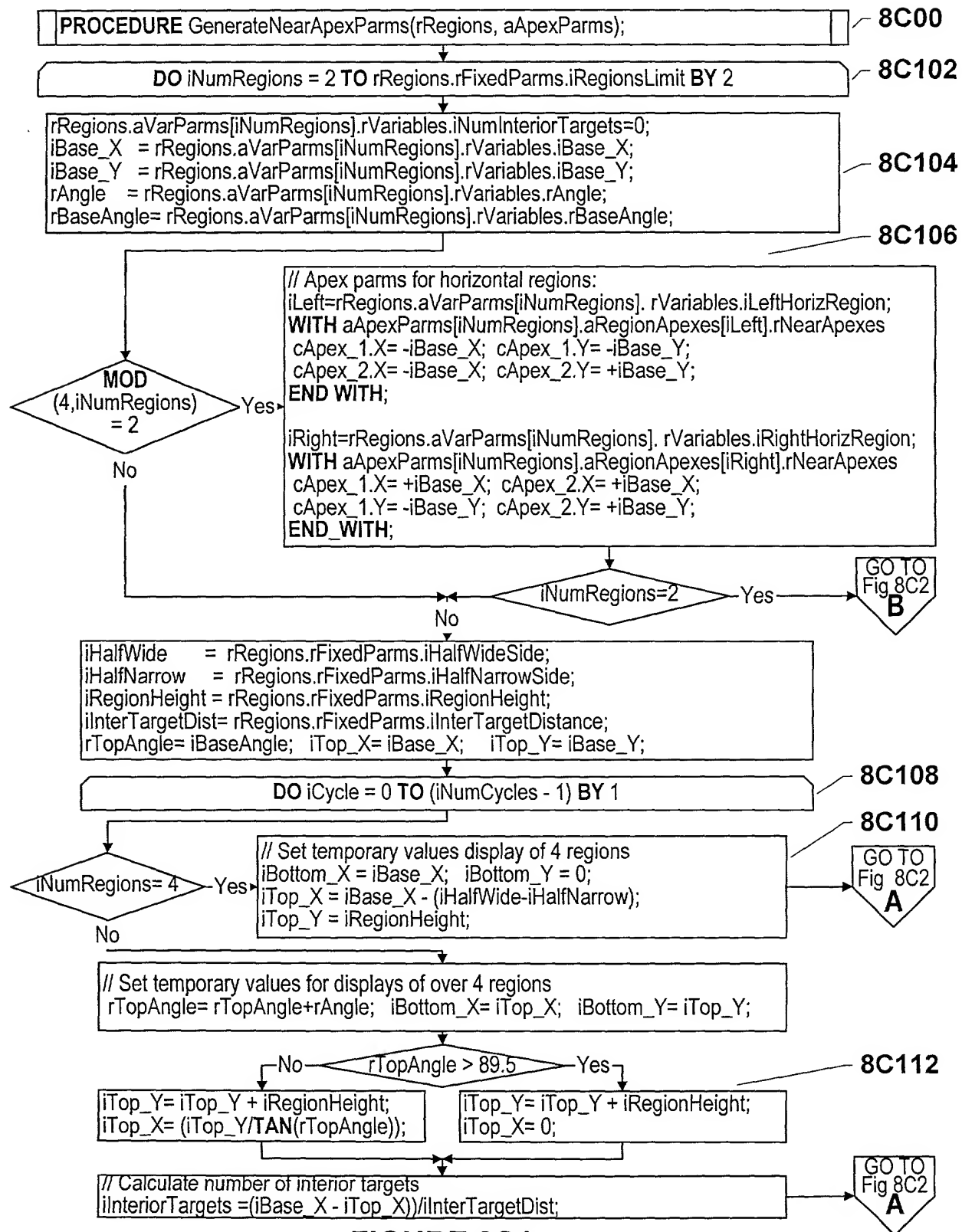


FIGURE 8B2

TEST 662007

38/68



39/68

8C202

A

```
WITH rRegions.aVarParms[iNumRegions].rVariables
iRow1 = iQRegion[1] - iCycle;
iRow2 = iQRegion[2] + iCycle;
iRow3 = iQRegion[3] - iCycle;
iRow4 = iQRegion[4] + iCycle;
END_WITH;
```

```
// Store near apex values of quadrant #1
WITH aApexParms[iNumRegions].aRegionApexes[iRow1].rNearApexes
iNumInteriorTargets = iInteriorTargets;
cApex1.X = -iBottom_X;
cApex1.Y = +iBottom_Y;
cApex2.X = -iTop_X;
cApex2.Y = +iTop_Y;
END_WITH;
```

```
// Store near apex values of quadrant #2
WITH aApexParms[iNumRegions].aRegionApexes[iRow2].rNearApexes
iNumInteriorTargets = iInteriorTargets;
cApex1.X = -iBottom_X;
cApex1.Y = -iBottom_Y;
cApex2.X = -iTop_X;
cApex2.Y = -iTop_Y;
END_WITH;
```

```
// Store near apex values of quadrant #3
WITH aApexParms[iNumRegions].aRegionApexes[iRow3].rNearApexes
iNumInteriorTargets = iInteriorTargets;
cApex1.X = +iBottom_X;
cApex1.Y = +iBottom_Y;
cApex2.X = +iTop_X;
cApex2.Y = +iTop_Y;
END_WITH;
```

```
// Store near apex values of quadrant #4
WITH aApexParms[iNumRegions].aRegionApexes[iRow4].rNearApexes
iNumInteriorTargets = iInteriorTargets;
cApex1.X = +iBottom_X;
cApex1.Y = -iBottom_Y;
cApex2.X = +iTop_X;
cApex2.Y = -iTop_Y;
END_WITH;
```

B

```
END_DO(Generation of NearApexParameters for given display);
```

```
END_DO(Generation of NearApexParameters for all displays);
```

RETURN

FIGURE 8C2

FIG. 8C2 "6662200"

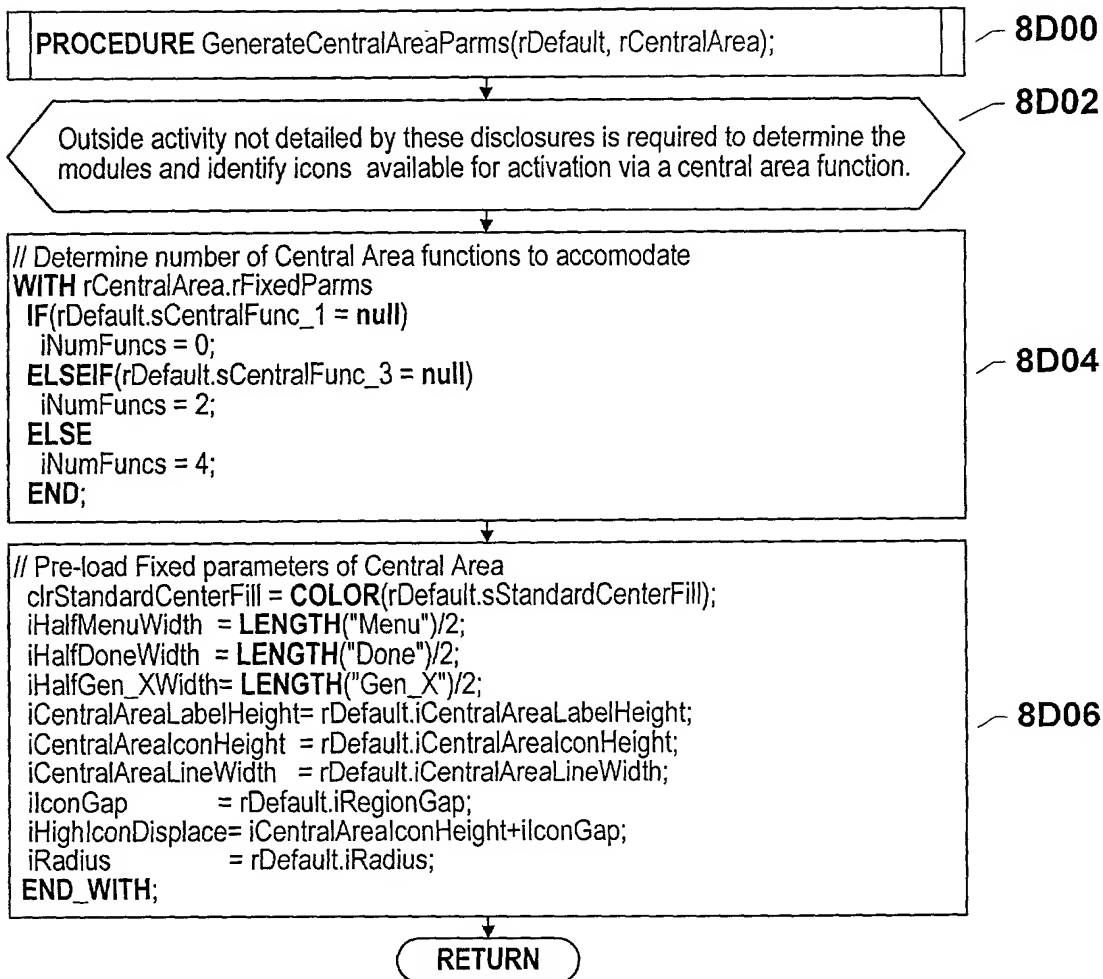
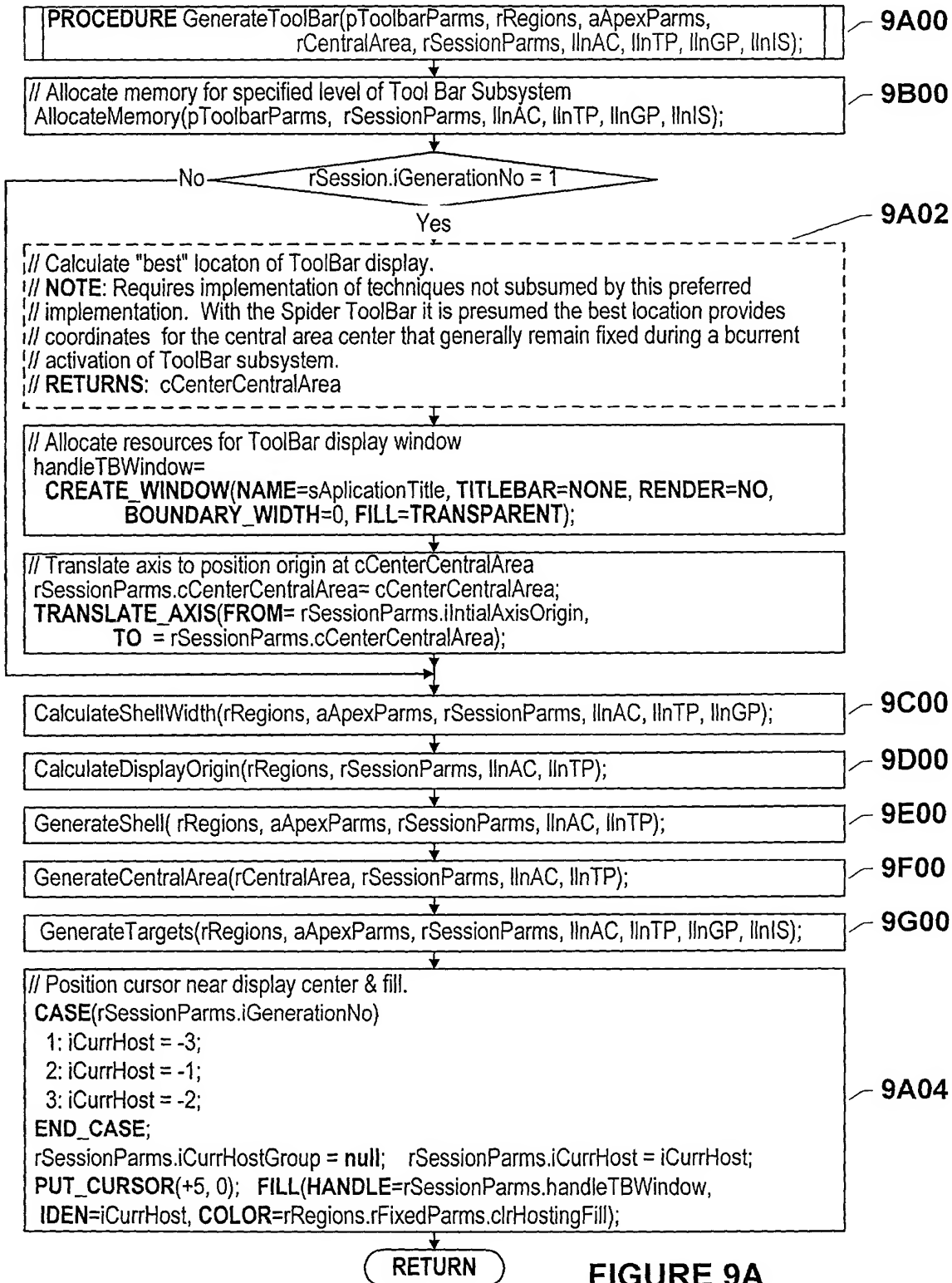


FIGURE 8D



42/68

PROCEDURE AllocateMemory(pToolBarParms,
rSessionParms, llnAC, llnTP, llnGP, llnIS);

9B00

// Acquire parameters of ToolBar list from system
iNumTargetGroups = GET_NUMBER_OF_ICON_GROUPS(*pToolBarParms);
iTotalNumTargets = GET_TOTAL_NUMBER_OF_ICONS(*pToolBarParms);
bDoneButton = GET_TRUE_IF_MULTI_SELECT(*pToolBarParms);
sToolBarLabel = GET_TOOLBAR_LABEL(*pToolBarParms);

9B02

// Allocate memory for a nAC node, load initial values.
iGenerationNo = rSessionParms.iGenerationNo + 1;
rSessionParms.iGenerationNo = iGenerationNo;
pnOldAC = rSessionParms.pnAC;
pnAC = GET_MEMORY(nAC); // memory => [2+3]
rSessionParms.pnAC = pnAC;
*pnAC.pChild = null;
*pnOldAC.pChild = pnAC;
IF(iGenerationNo = 1)
*pnAC.pParent = null;
rSessionParms.plnInitialAC = pnAC;
ELSE *pnAC.pParent = pnOldAC;
END;

9B04

// Allocate memory for a nTP node and load initial values.
pnTP = GET_MEMORY(nTP); // memory => [[1+7+2*2]
*pnAC.rActivityControl.pnTP = pnTP;
IF(iGenerationNo = 1) *pnTP.pParent = null;
ELSE *pnTP.pParent = *pnOldAC.rActivityControl.pnTP;
END;
*pnTP.rToolBarParms.iNumRegions = ((iNumTargetGroups+1)/2)*2;
*pnTP.rToolBarParms.iNumTargetGroups = iNumTargetGroups;
*pnTP.rToolBarParms.iTotalNumTargets = iTotalNumTargets;
*pnTP.rToolBarParms.bDoneButton = bMultiSelect;
*pnTO.rToolBarParms.sToolBarLabel = sToolBarLabel;

9B06

// Allocate memory for a nGP node & load initial values.
pnGP = GET_MEMORY(nGP); // memory =>
*pnAC.rActivityControl.pnGP = pnGP; // 1+4*iNumTargetGroups
IF(iGenerationNo = 1) *pnGP.pParent = null;
ELSE *pnGP.pParent = *pnOldAC.rActivityControl.pnGP;
END;
CONVERT(ICON_GROUPS_OF(*pToolBarParms)
TO_FORMAT_OF(rGroupParms) AND_LOAD_TO(*pnGP.aTargetGroups));

9B08

// Allocate memory for a nIS node & load initial values.
pnIS = GET_MEMORY(nIS); // memory =>
*pnAC.rActivityControl.pnIS = pnIS; // [1+6*iTotalNumTargets]
IF(iGenerationNo = 1) *pnIS.pParent = null;
ELSE *pnIS.pParent = *pnOldAC.rActivityControl.pnIS;
END;
CONVERT(ICONS_OF(*pToolBarParms)
TO_FORMAT_OF(rIconParms) AND_LOAD_TO(*pnIS.alconSet));

9B10

RETURN

FIGURE 9B

FIGURE 9B

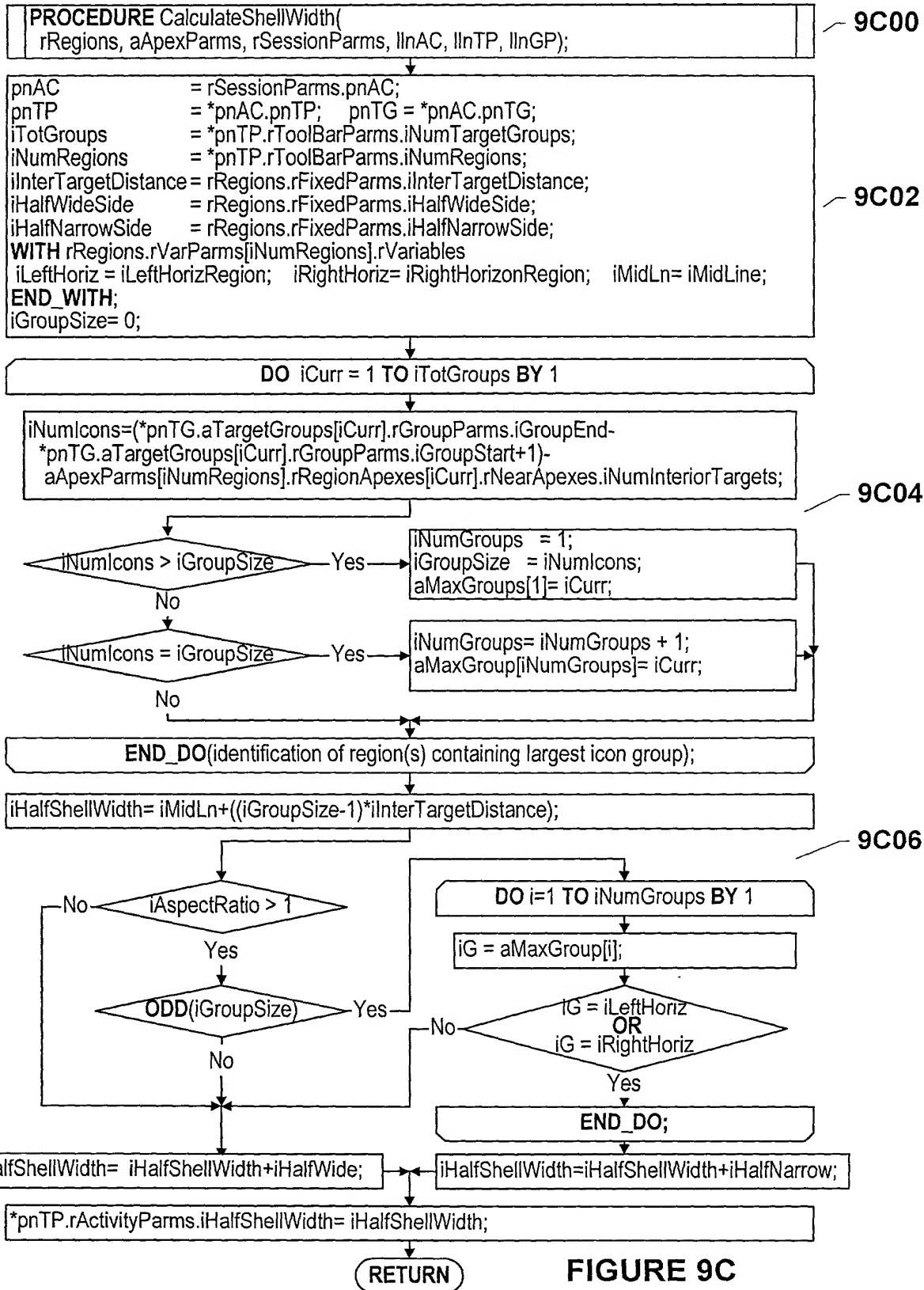


FIGURE 9C

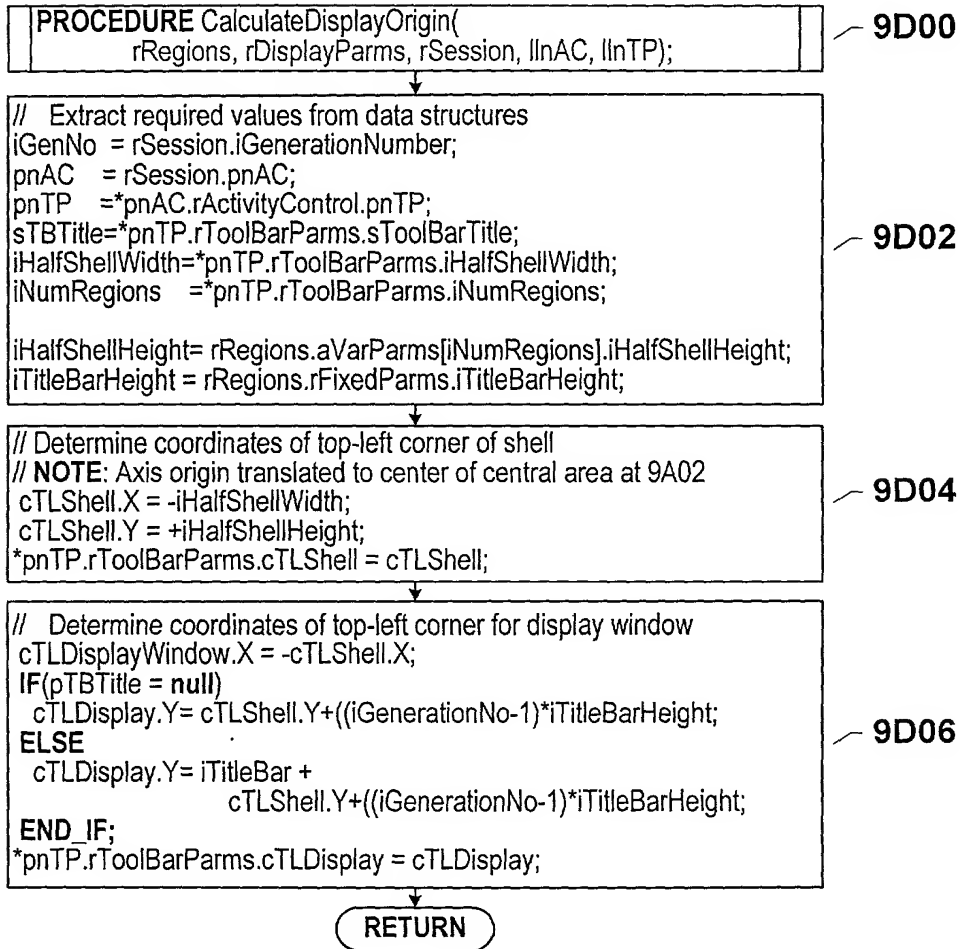
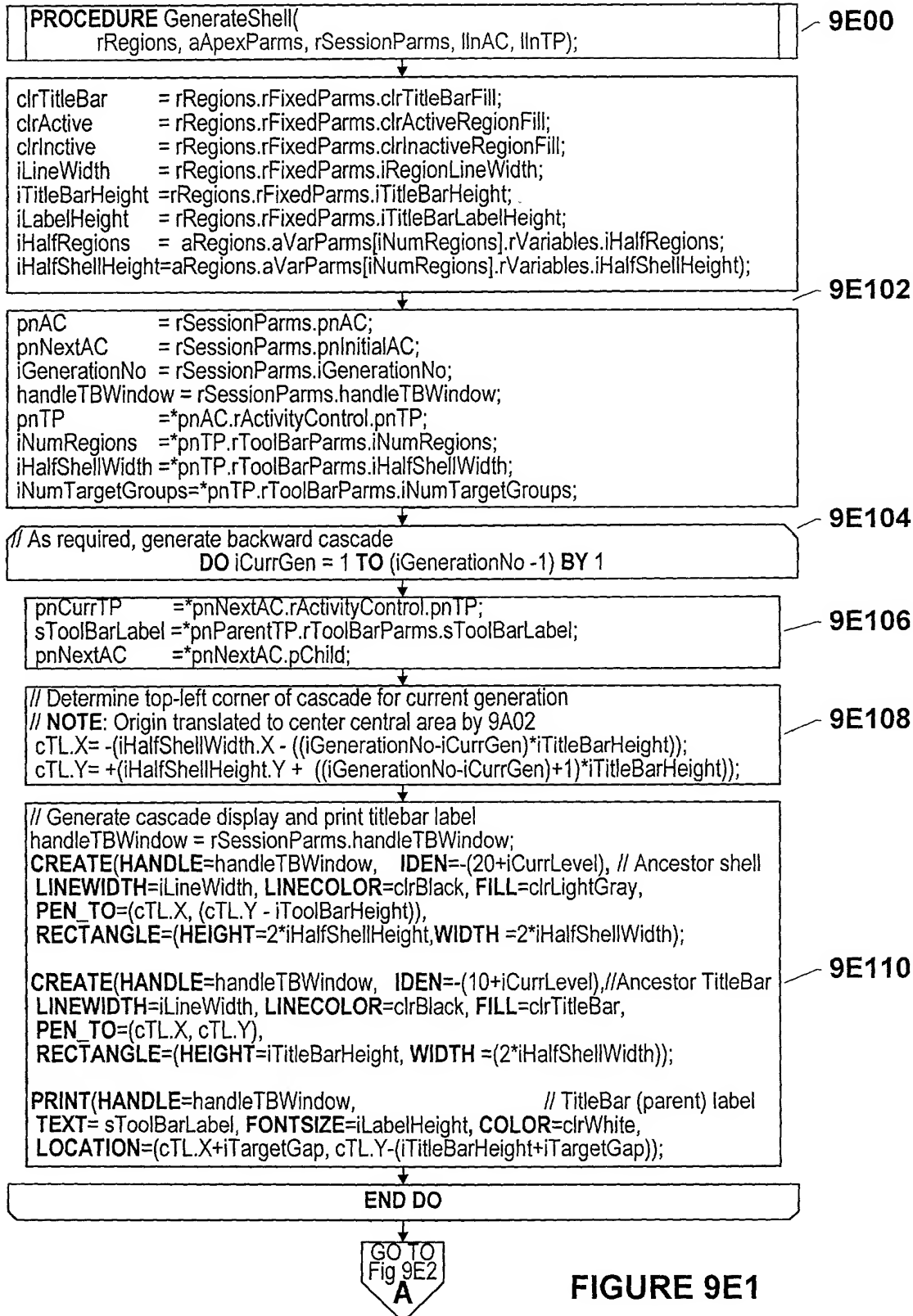
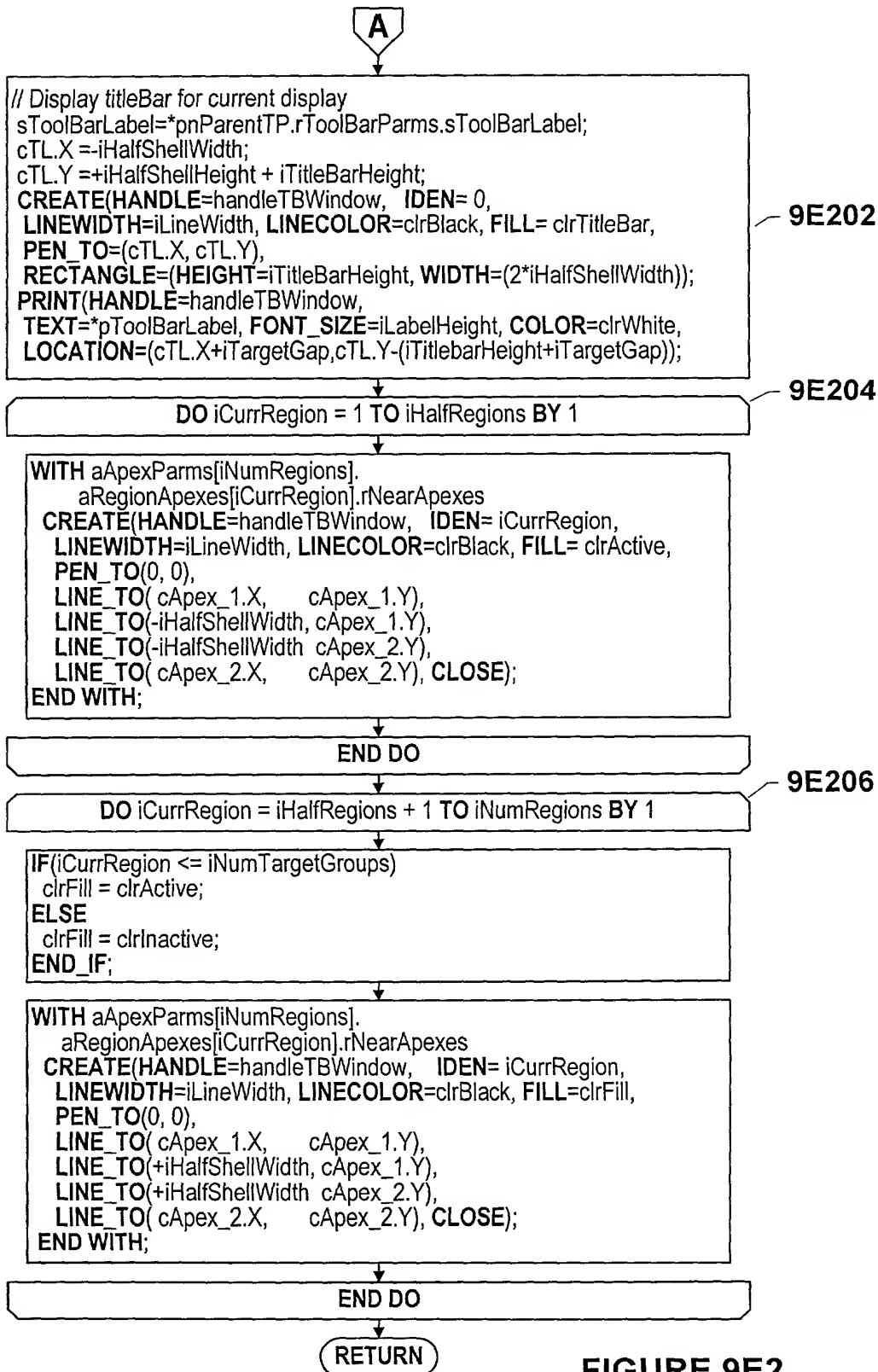


FIGURE 9D





47/68

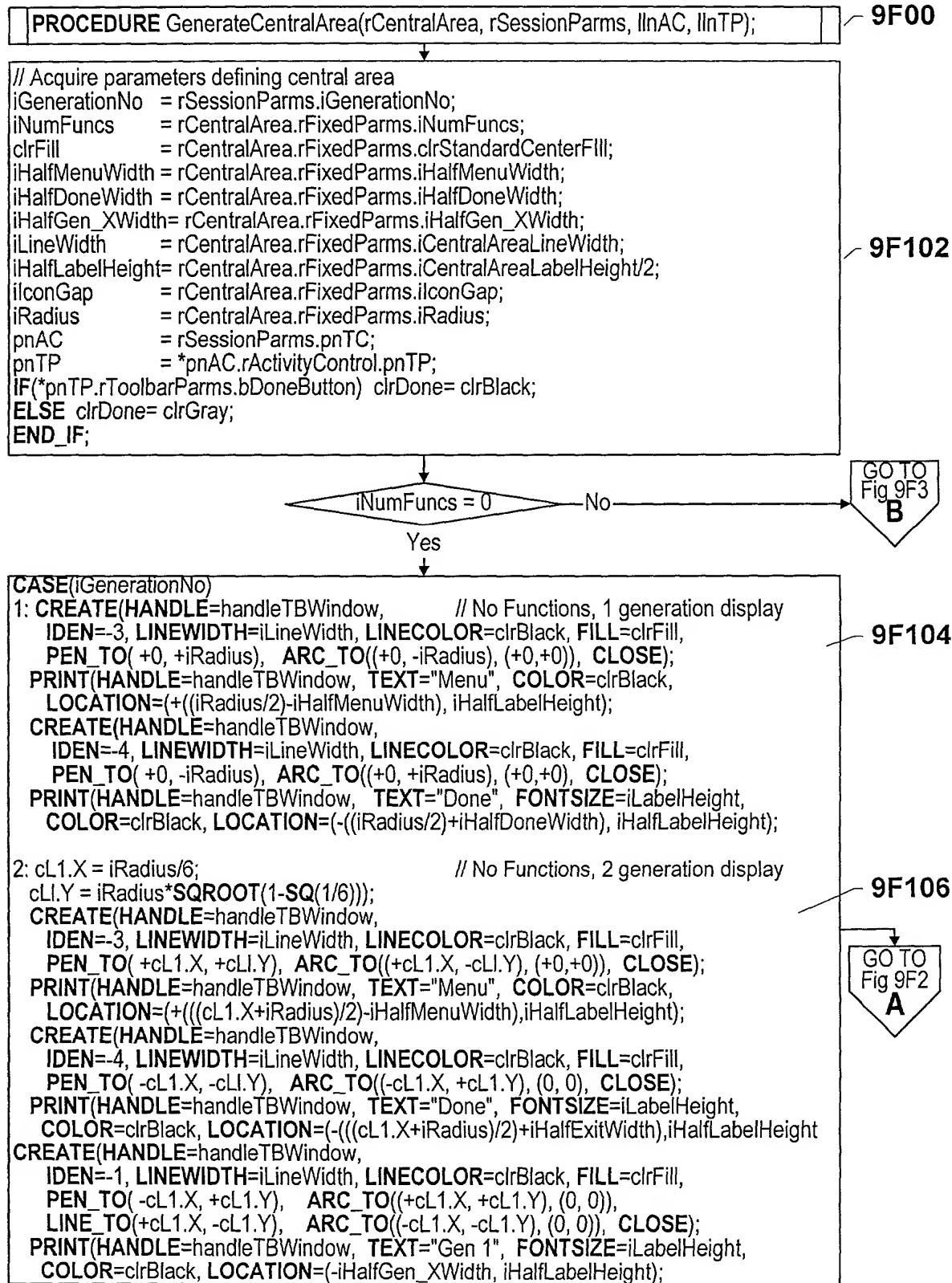


FIGURE 9F1

FIG. 9F1

48/68

A

9F202

```
3: cL2.X = iRadius/2; // No Functions, 3 generation display
   cL2.Y = iRadius*SQROOT(1-SQ(1/2));

CREATE(HANDLE=handleTBWindow,
  IDEN=-3, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
  PEN_TO( +cL2.X, +cL2.Y),
  ARC_TO((+cL2.X, -cL2.Y), (+0,+0)), CLOSE);
PRINT(HANDLE=handleTBWindow,
  TEXT="Menu", FONTSIZE=iLabelHeight, COLOR=clrBlack,
  LOCATION=(+(((cL2.X+iRadius)/2)-iHalfMenuWidth),iHalfLabelHeight);

CREATE(HANDLE=handleTBWindow,
  IDEN=-4, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
  PEN_TO(-cL2.X, -cL2.Y),
  ARC_TO((-cL2.X, +cL2.Y), (0, 0), CLOSE);
PRINT(HANDLE=handleTBWindow,
  TEXT="Done", FONTSIZE=iLabelHeight, COLOR=clrDone,
  LOCATION=(-(((cL2.X+iRadius)/2)+iHalfDoneWidth),iHalfLabelHeight);

CREATE(HANDLE=handleTBWindow,
  IDEN=-1, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
  PEN_TO( 0, +iRadius),
  ARC_TO((+cL2.X, +cL2.Y), (0, 0)),
  LINE_TO(+cL2.X, -cL2.Y),
  ARC_TO(( 0, -iRadius), (0, 0)), CLOSE);
PRINT(HANDLE=handleTBWindow,
  TEXT="Gen 1", FONTSIZE=iLabelHeight, COLOR=clrBlack,
  LOCATION=(+cL2.X/2 - iHalfGen_XWidth), iHalfLabelHeight);

CREATE(HANDLE=handleTBWindow,
  IDEN=-2, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
  PEN_TO( 0, -iRadius),
  ARC_TO((-cL2.X, -cL2.Y), (0, 0)),
  LINE_TO(-cL2.X, +cL2.Y),
  ARC_TO(( 0, +iRadius), (0, 0)), CLOSE);
PRINT(HANDLE=handleTBWindow,
  TEXT="Gen 2", FONTSIZE=iLabelHeight, COLOR=clrBlack,
  LOCATION=(-cL2.X/2 + iHalfGen_XWidth), iHalfLabelHeight);
```

RETURN

FIGURE 9F2

FIGURE 9F2

B

```
ilconWidth = rCentralArea.rFixedParms.iCentralArealconHeight;
iHalfilconWidth=(ilconWidth)/2;
ilconDisplace = rCentralArea.rFixedParms.iHighIconDisplace;
plcon_1 = rCentralArea.aFuncs[3].rLocators.plcon;
plcon_2 = rCentralArea.aFuncs[4].rLocators.plcon;
plcon_3 = rCentralArea.aFuncs[5].rLocators.plcon;
plcon_4 = rCentralArea.aFuncs[6].rLocators.plcon;
cL1.Y = iHalfLineWidth + iHalfLabelHeight + ilconGap;
cL1.X = SQROOT(SQ(iRadius) - SQ(cL1.Y));
cL2.X = iRadius/6; cL3.X = iRadius/2;
```

9F302

No
iNumFuncs = 2
Yes

9F304

```
CREATE(HANDLE=handleTBWindow, // Two central area functions
IDEN=-5, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO(-cL1.X, +cL1.Y), ARC_TO(+cL1.X, +cL1.Y), (+0,+0)), CLOSE);
PAINT(HANDLE=handleTBWindow,
ICON=*plcon_1, LOCATION=(-iHalfilconWidth, +(cL1.Y+ilconDisplace)));

CREATE(HANDLE=handleTBWindow,
IDEN=-6, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO(+cL1.X, -cL1.Y), ARC_TO(-cL1.X, -cL1.Y), (+0,+0), CLOSE);
PAINT(HANDLE=handleTBWindow,
ICON=*plcon_2, LOCATION=(-iHalfilconWidth, -(cL1.Y+ilconGap)));
```

9F306

```
CREATE(HANDLE=handleTBWindow, // Four central area functions
IDEN=-5, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO(-cL1.X, +cL1.Y), ARC_TO(0, iRadius), (+0,+0)), LINE_TO(0, +cL1.Y), CLOSE);
PAINT(HANDLE=handleTBWindow,
ICON=*plcon_1, LOCATION=(-iHighDisplace, +(cL1.Y+ilconDisplace)));

CREATE(HANDLE=handleTBWindow,
IDEN=-7, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO(0, +iRadius), ARC_TO(+cL1.X, +cL1.Y), (+0,+0)), LINE_TO(0, +cL1.Y), CLOSE);
PAINT(HANDLE=handleTBWindow,
ICON=*plcon_3, LOCATION=(+ilconGap, +(cL1.Y+ilconDisplace)));

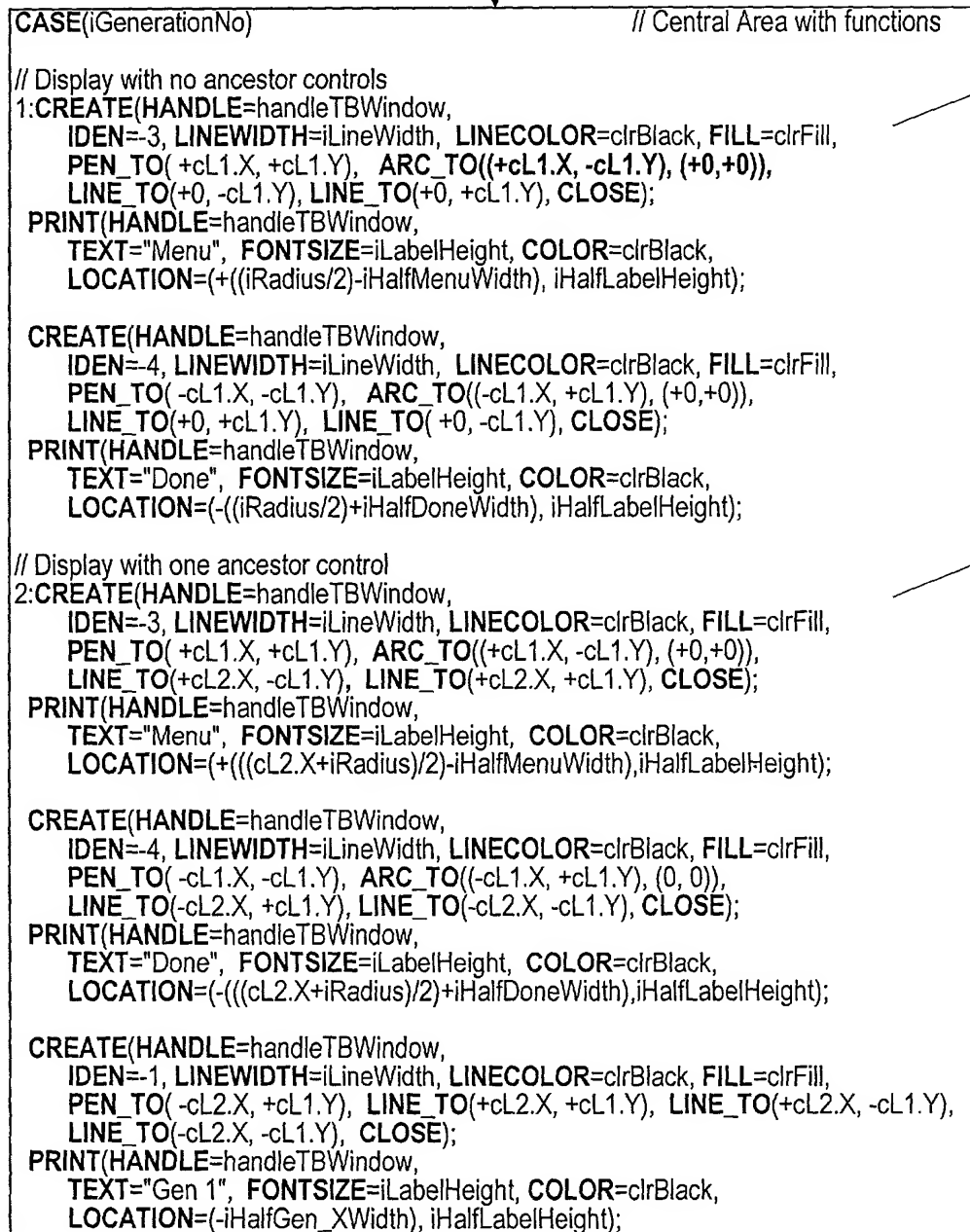
CREATE(HANDLE=handleTBWindow,
IDEN=-6, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO(+cL1.X, -cL1.Y), ARC_TO((0, -iRadius), (+0, +0)), LINE_TO(0, -cL1.Y), CLOSE);
PAINT(HANDLE=handleTBWindow,
ICON=*plcon_2, LOCATION=(-iHighDisplace, -(cL1.Y+ilconGap)));

CREATE(HANDLE=handleTBWindow,
IDEN=-8, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO(0, -iRadius), ARC_TO((-cL1.X, -cL1.Y), (0, 0)), LINE_TO(0, -cL1.Y), CLOSE);
PAINT(HANDLE=handleTBWindow,
ICON=*plcon_4, LOCATION=(+ilconGap, -(cL1.Y+ilconGap)));
```

GOTO
Fig 9F4
C

FIGURE 9F3

100299-12301



9F402

9F404



FIGURE 9F4

TEST "66622001"

D

9F502

```
// Display with two ancestor controls
3:CREATE(HANDLE=handleTBWindow,
IDEN=-3, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO( +cL1.X, +cL1.Y),
ARC_TO((+cL1.X, -cL1.Y), (+0,+0)),
LINE_TO(+cL3.X, -cL1.Y),
LINE_TO(+cL3.X, +cL1.Y), CLOSE);
PRINT(HANDLE=handleTBWindow,
TEXT="Menu", FONTSIZE=iLabelHeight, COLOR=clrBlack,
LOCATION=(+(((iRadius+cL3)/2)-iHalfMenuWidth), +iHalfLabelHeight);

CREATE(HANDLE=handleTBWindow,
IDEN=-4, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO( -cL1.X, -cL1.Y),
ARC_TO((-cL1.X, +cL1.Y), (0, 0),
LINE_TO(-cL3.X, +cL1.Y),
LINE_TO(-cL3.X, -cL1.Y), CLOSE);
PRINT(HANDLE=handleTBWindow,
TEXT="Done", FONTSIZE=iLabelHeight, COLOR=clrBlack,
LOCATION=(-(((iRadius+cL3)/2)+iHalfDoneWidth), +iHalfLabelHeight);

CREATE(HANDLE=handleTBWindow,
IDEN=-1, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO( -cL3.X, +cL1.Y),
LINE_TO(+0, +cL1.Y),
LINE_TO(+0, -cL1.Y),
LINE_TO(-cL3.X, -cL1.Y), CLOSE);
PRINT(HANDLE=handleTBWindow,
TEXT="Gen 1", FONTSIZE=iLabelHeight, COLOR=clrBlack,
LOCATION=(-((cL3/2)+iHalfGen_XWidth), +iHalfLabelHeight);

CREATE(HANDLE=handleTBWindow,
IDEN=-2, LINEWIDTH=iLineWidth, LINECOLOR=clrBlack, FILL=clrFill,
PEN_TO( +0, +cL1.Y),
LINE_TO(+cL3.X, +cL1.Y),
LINE_TO(+cL3.X, -cL1.Y),
LINE_TO(+0, -cL1.Y), CLOSE);
PRINT(HANDLE=handleTBWindow,
TEXT="Gen 2", FONTSIZE=iLabelHeight, COLOR=clrBlack,
LOCATION=(+((cL3/2)-iHalfGen_XWidth), +iHalfLabelHeight);

END_CASE;
```

RETURN

FIGURE 9F5

FOOT " 55622007

PROCEDURE GenerateTargets(
rRegions, aApexParms, rSessionParms, llnAC, llnTP, llnGP, llnIS);

9G00

// Acquire Fixed Region parameters
iAspectRatio = rRegions.rFixedParms.iAspectRatio;
iHalfIconWidth = rRegions.rFixedParms.iShellIconHeight/2;
iBaseHalfNarrow = rRegions.rFixedParms.iHalfNarrowSide;
iBaseHalfWide = rRegions.rFixedParms.iHalfWideSide;
iBaseInterTarget = rRegions.rFixedParms.iInterTargetDistance;
iLabelDisplace = rRegions.rFixedParms.iLowLabelDisplace_Y;
iIconDisplace = rRegions.rFixedParms.iLowIconDisplace_Y;
iHighDisplace = rRegions.rFixedParms.iHighDisplace_Y;

9G102

// Acquire Variable Region parameters
pnAC = rSessionParms.pnAC;
pnTP = *pnAC.rActivityControl.pnTP;
pnGP = *pnAC.rActivityControl.pnGP;
iNumRegions = *pnTP.rToolBarParms.iNumRegions;
iNumTargetGroups = *pnTP.rToolBarParms.iNumTargetGroups;
WITH rRegions.aVarParms[iNumRegions].Variables
iTargetDisplace = iTargetDisplace_Y; iHalfRegion = iHalfRegions;
iLHR = iLeftHorizontalRegion; iRHR = iRightHorizontalRegion;
iBaseX = iBase_X; iBaseMidLine = iMidLine;
iQRegion_1 = aQRegion[1]; iQRegion_2 = aQRegion[2];
iQRegion_3 = aQRegion[3]; iQRegion_4 = aQRegion[4];
END_WITH;

9G104

DO iCurrRegion = 1 **TO** iNumTargetGroups **BY** 1

9G106

// Use near apexes of current region to determine apexes
WITH aApexParms[iNumRegions].
aRegionApexes[iCurrRegion].rNearApexes
iNumInnerTargets = iNumInteriorTargets;
c1.Y = cApex_1.Y; c2.Y = cApex_2.Y;
c3.Y = c2.Y; c4.Y = c1.Y;
c1.X = cApex_1.X; c2.X = cApex_2.X;
END_WITH;

9G108

// Set locational values for each display half:
IF (iCurrRegion > iHalfRegion)
iHalfNarrow = +iBaseHalfNarrow; iHalfWide = +iBaseHalfWide;
iInterTargetDistance = +iBaseInterTarget;
iCurrBase_X = +iBaseX;
iCurrMidLine = +iBaseMidLine;
ELSE
iHalfNarrow = -iBaseHalfNarrow; iHalfWide = -iBaseHalfWide;
iInterTargetDistance = -iBaseInterTargetDistance;
iCurrBase_X = -iBaseX;
iCurrMidLine = -iBaseMidLine;
END_IF;

9G110



FIGURE 9G1

FIG. 9G1 OF 52

53/68

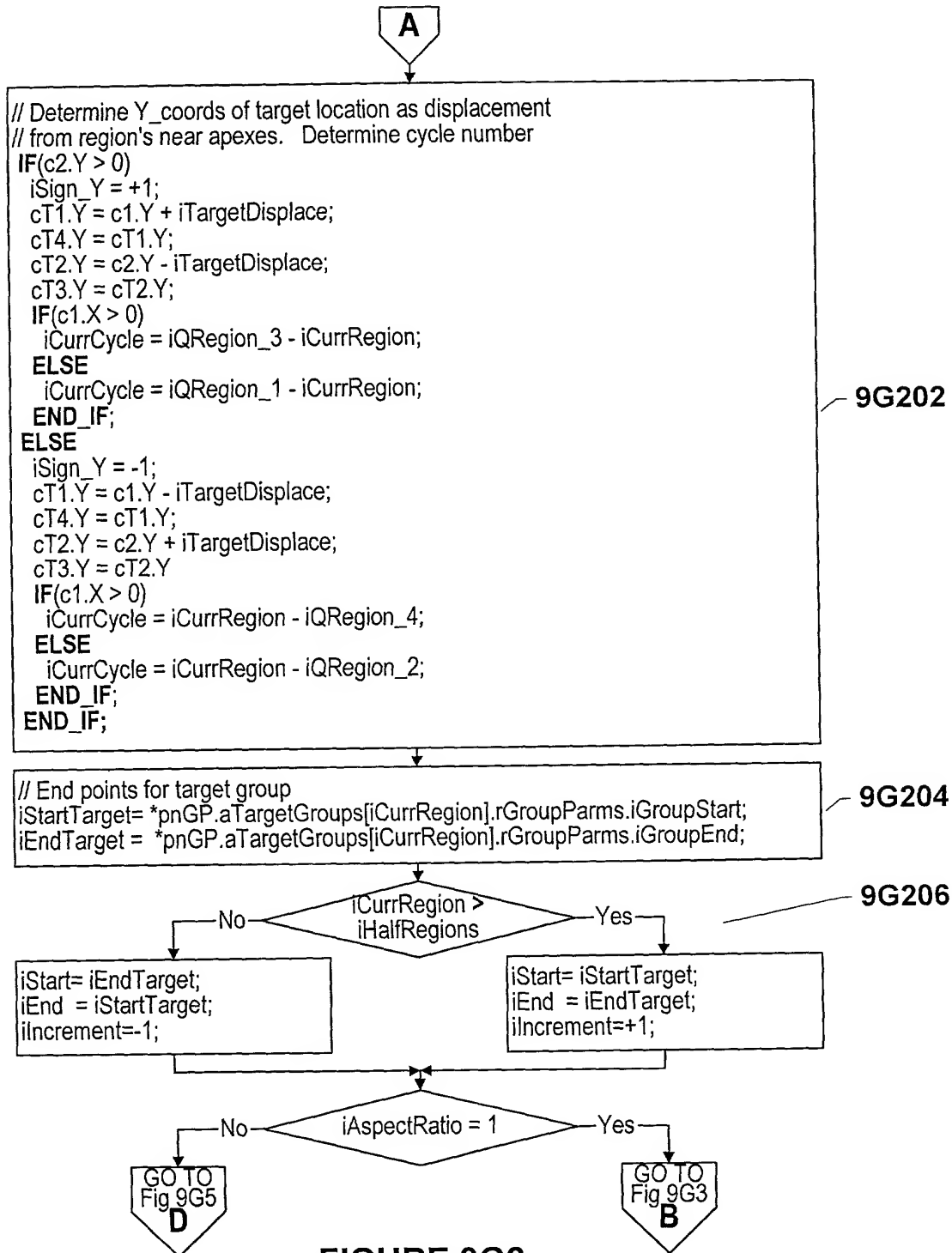
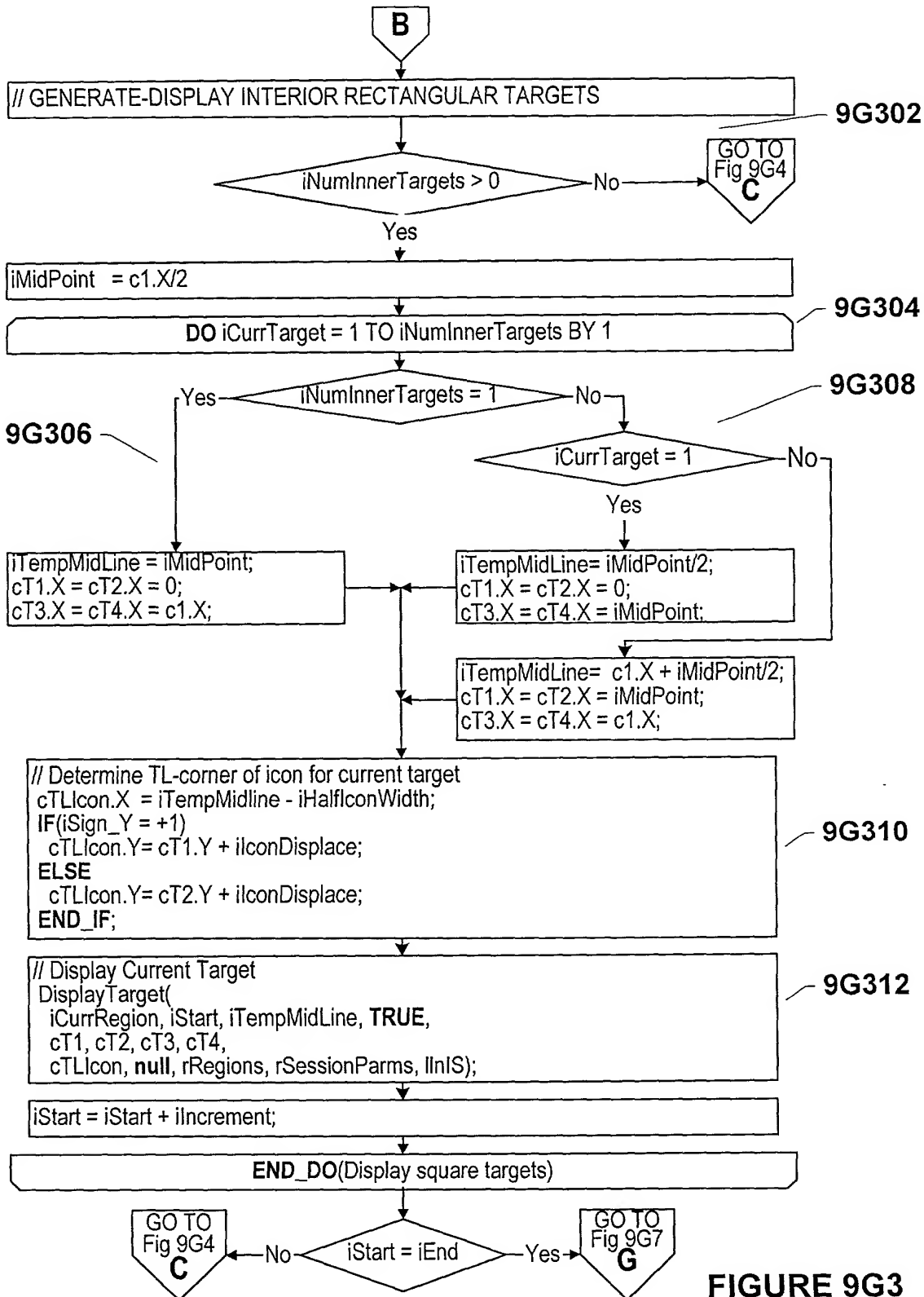
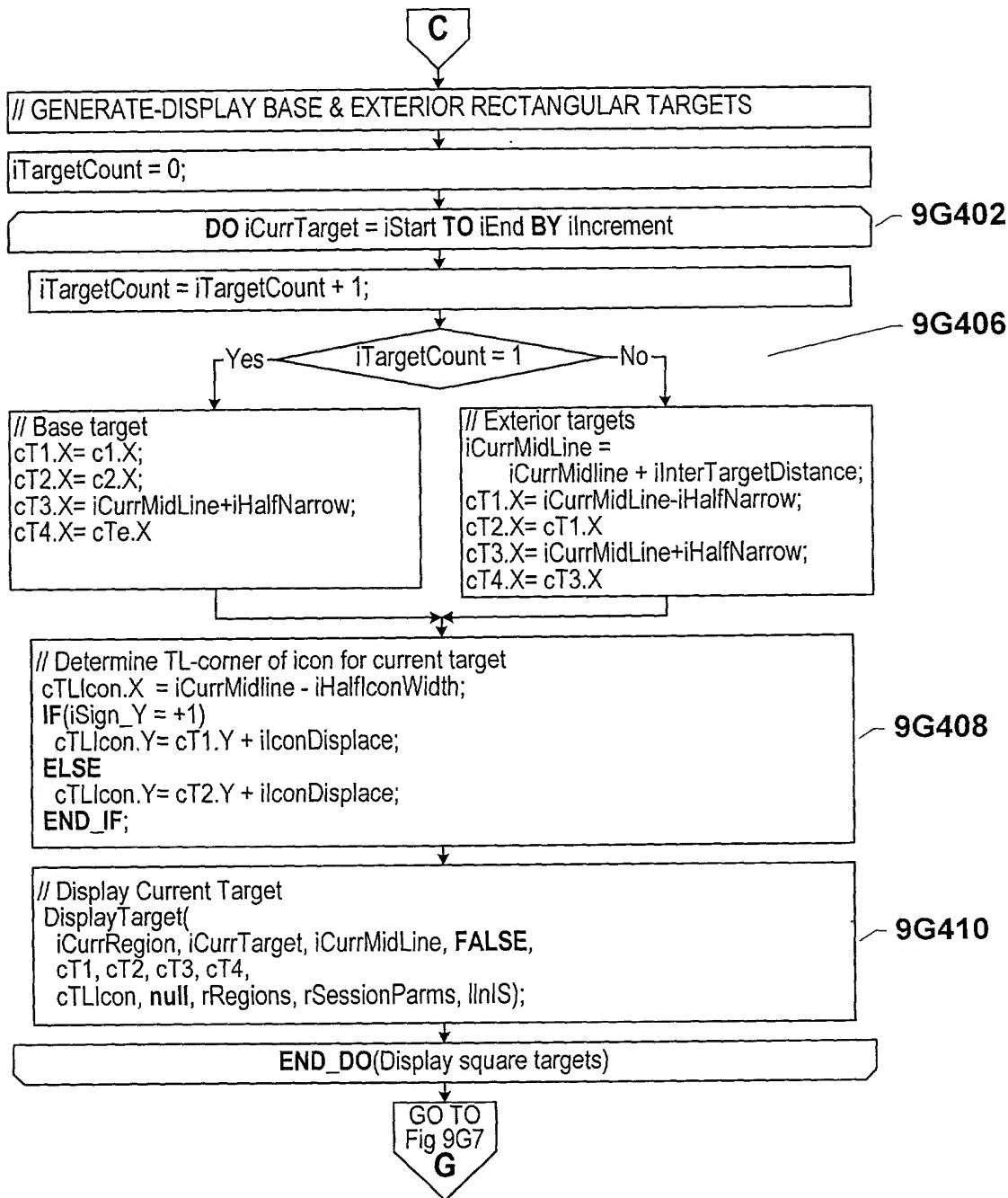
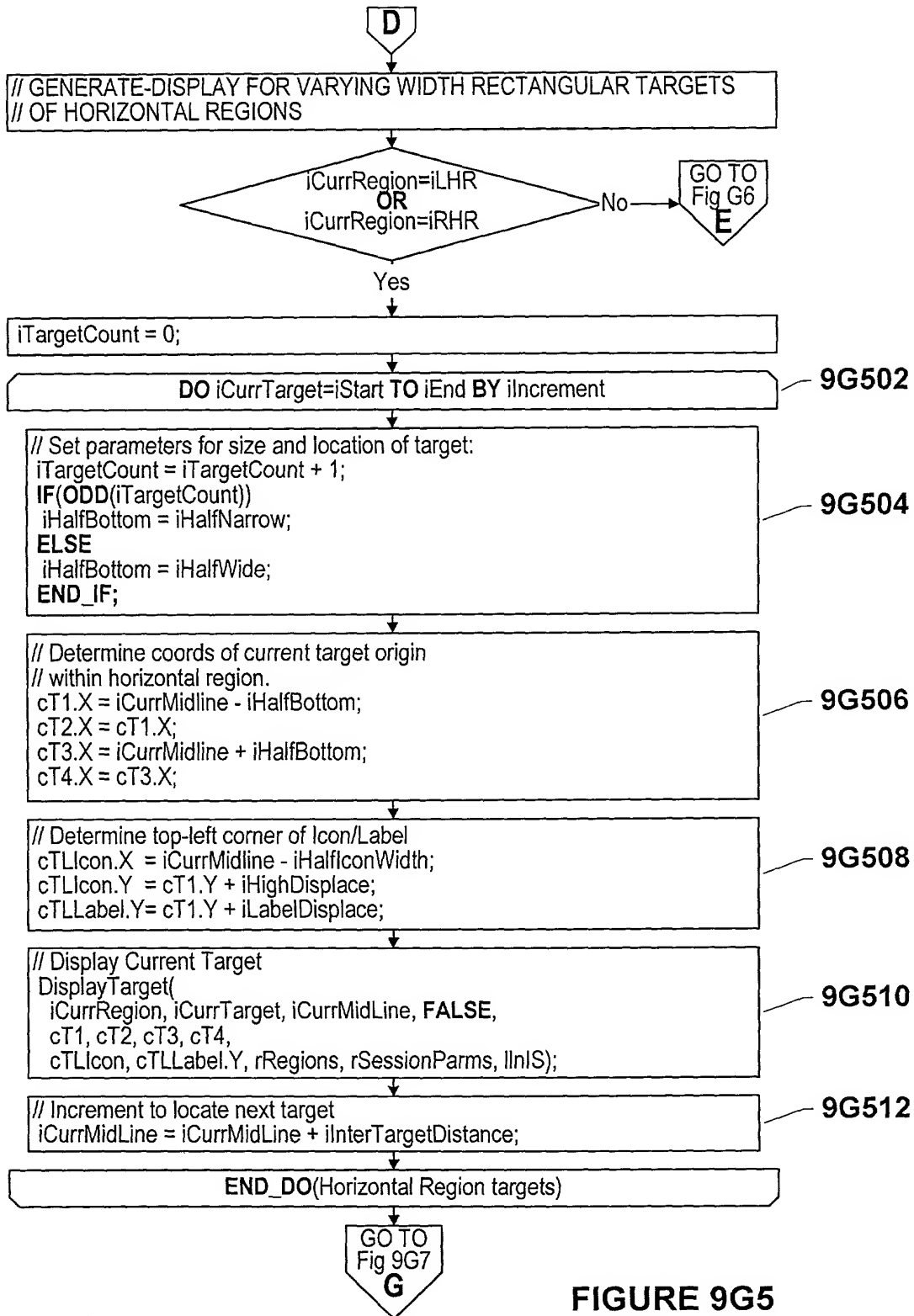


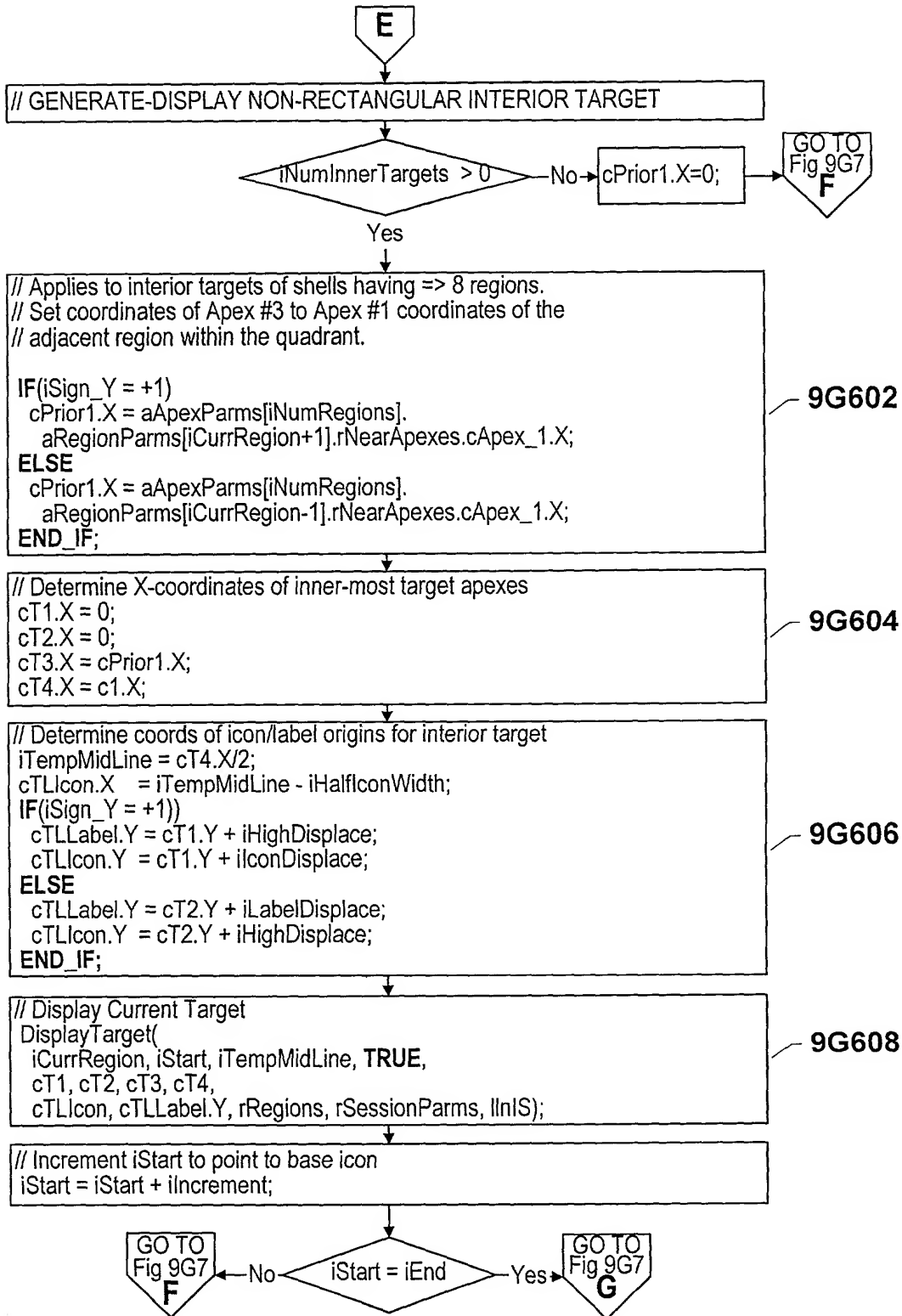
FIGURE 9G2

54/68







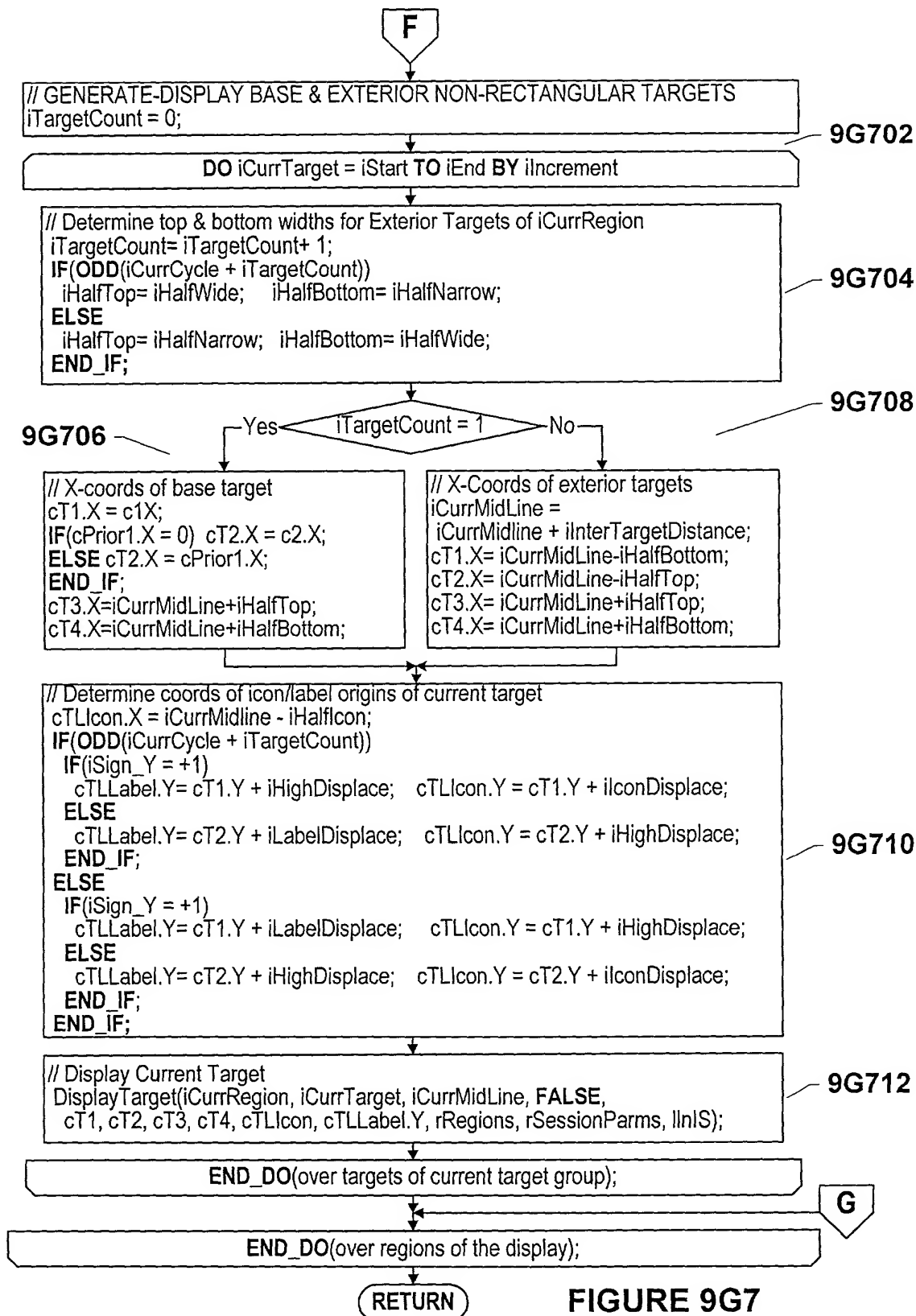


9G602

9G604

9G606

9G608



59/68

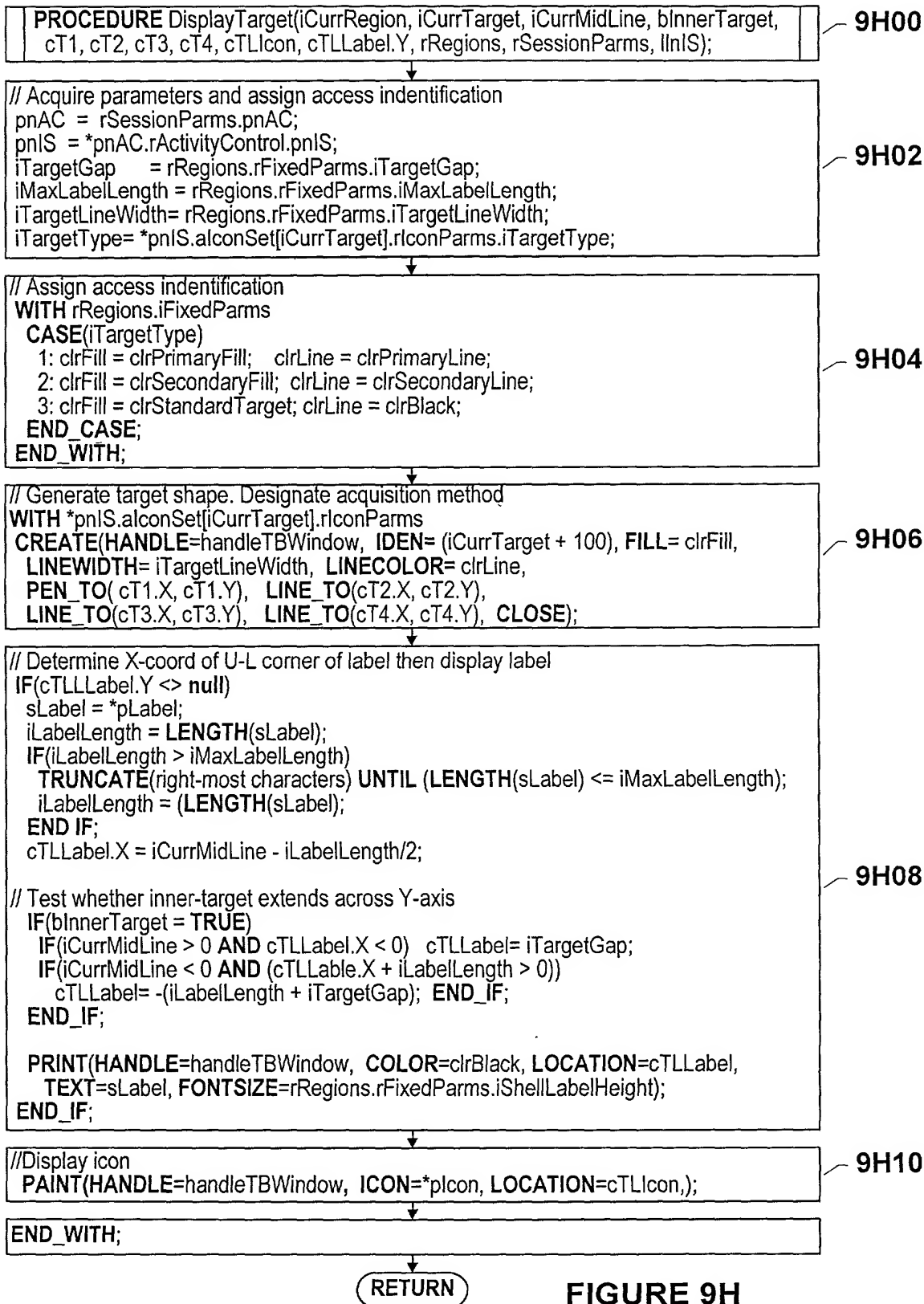


FIGURE 9H

FIGURE 9H

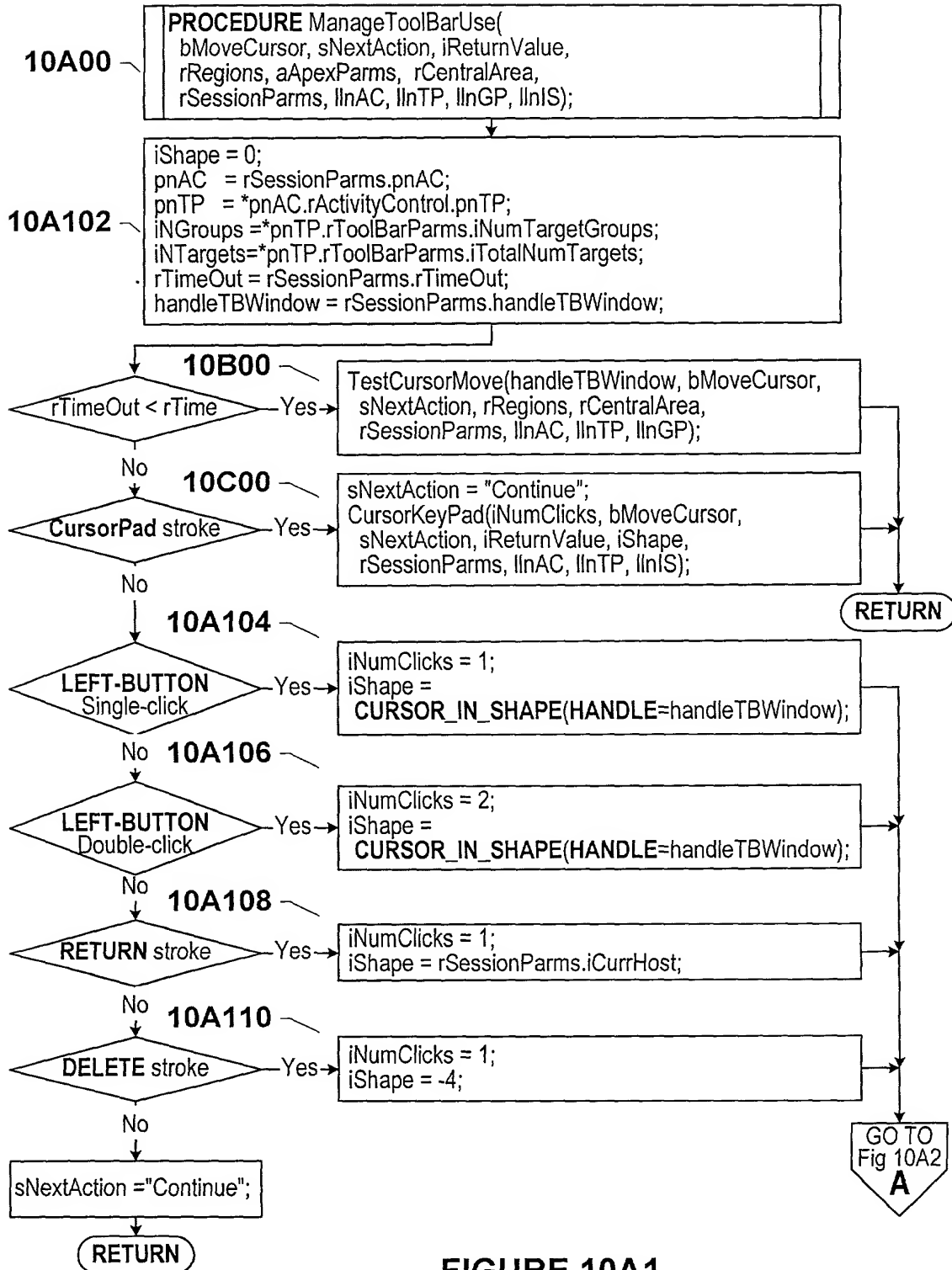
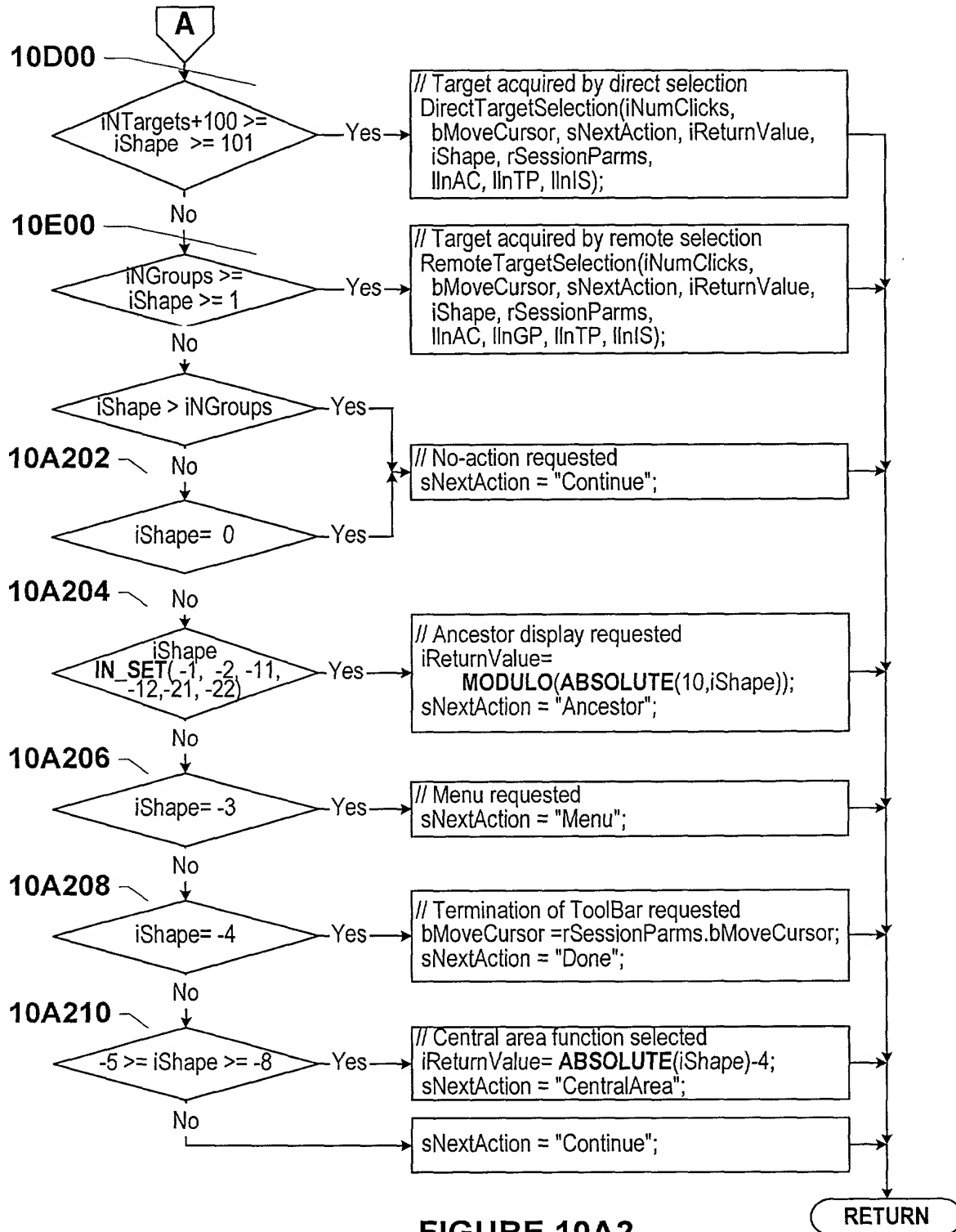


FIGURE 10A1



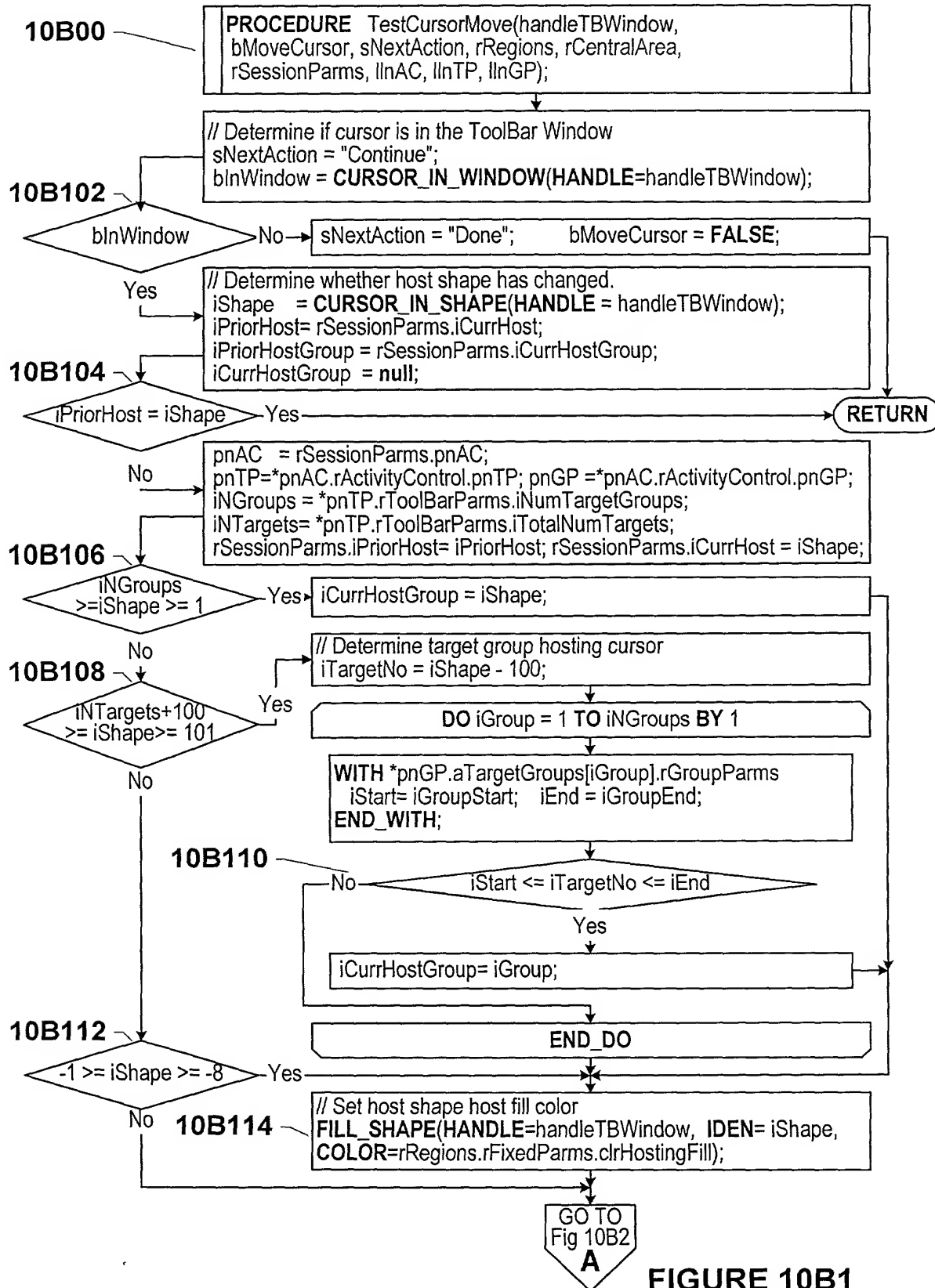


FIGURE 10B1

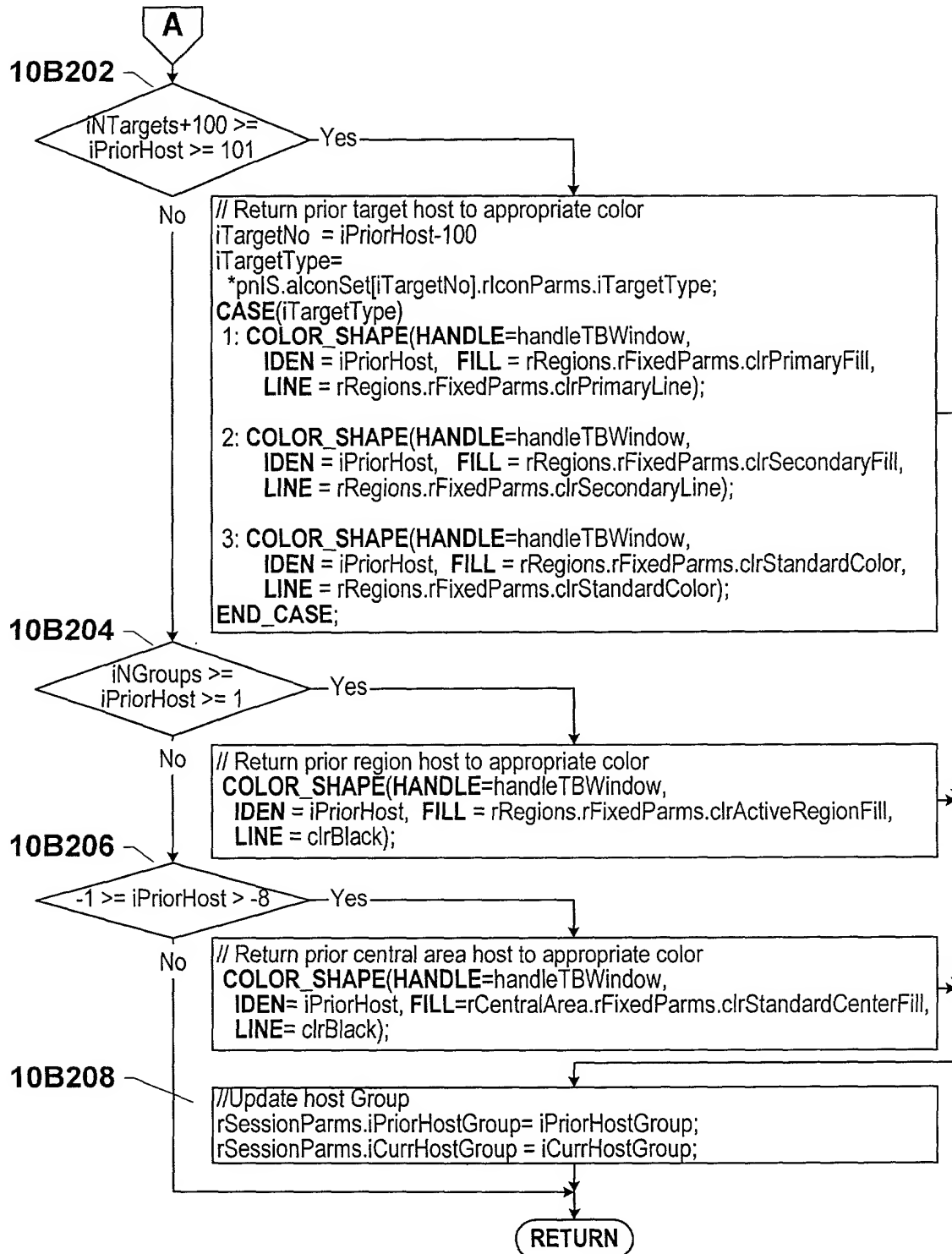


FIGURE 10B2

PROCEDURE CursorKeyPad(
iNumClicks, bMoveCursor, sNextAction, iReturnValue, iShape,
rSessionParms, llnAC, llnTP, llnIS);

This invention subsumes the availability of cursor manipulation via the cursor keypad. The general approach recommended for keypad manipulation of the cursor for a Spider format is disclosed for the Spider Combo Box by patent number 6,239,803. Variations of keypad manipulation appropriate for Spider Toolbar manipulation will be apparent to the person of normal skills in these arts and will not be reviewed by this implementation.

10C00

RETURN

FIGURE 10C

PROCEDURE DirectTargetSelection(iNumClicks,
bMoveCursor, sNextAction, iReturnValue, iShape,
rSessionParms, llnAC, llnTP, llnIS);

10D00

10D02

10D04

iNumClicks = 1

No

Yes

sNextAction= "Continue";

// Access required values
pnAC = rSessionParms.pnAC;
pnTP = *pnAC.rActivityControl.pnTP;
pnIS = *pnAC.rActivityControl.pnIS;
iTargetNo= iShape - 100;
pChildToolBar= *pnIS.alconSet[iTargetNo].rlconParms.pChildToolBar;
pOtherControl= *pnIS.alconSet[iTargetNo].rlconParms.pOtherControl;
iMenuTag= *pnIS.alconSet[iTargetNo].rlconParms.iMenuTag;
*pnTP.rToolBarParms.pSelectedTargetIden = iTargetNo;

pChildToolBar = null

No

Yes

// Sub-Toolbar requested
sNextAction = "Toolbar";
iReturnValue = pChildToolBar;

pOtherControl = null

No

Yes

// Sub-Toolbar requested
sNextAction = "Control";
iReturnValue = pOtherControl;

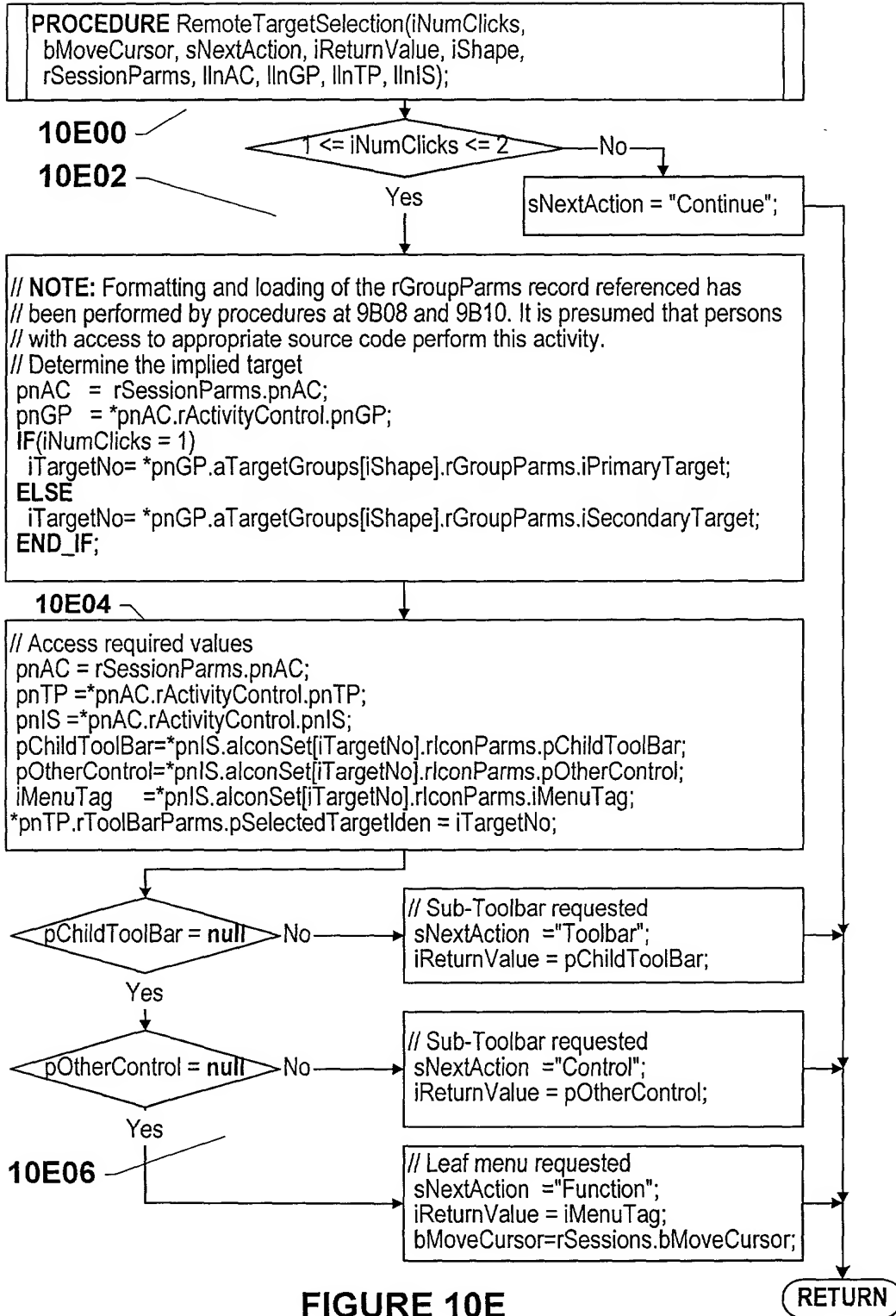
10D06

// Leaf menu requested
sNextAction = "Function";
iReturnValue = iMenuTag;
bMoveCursor = rSessions.bMoveCursor;

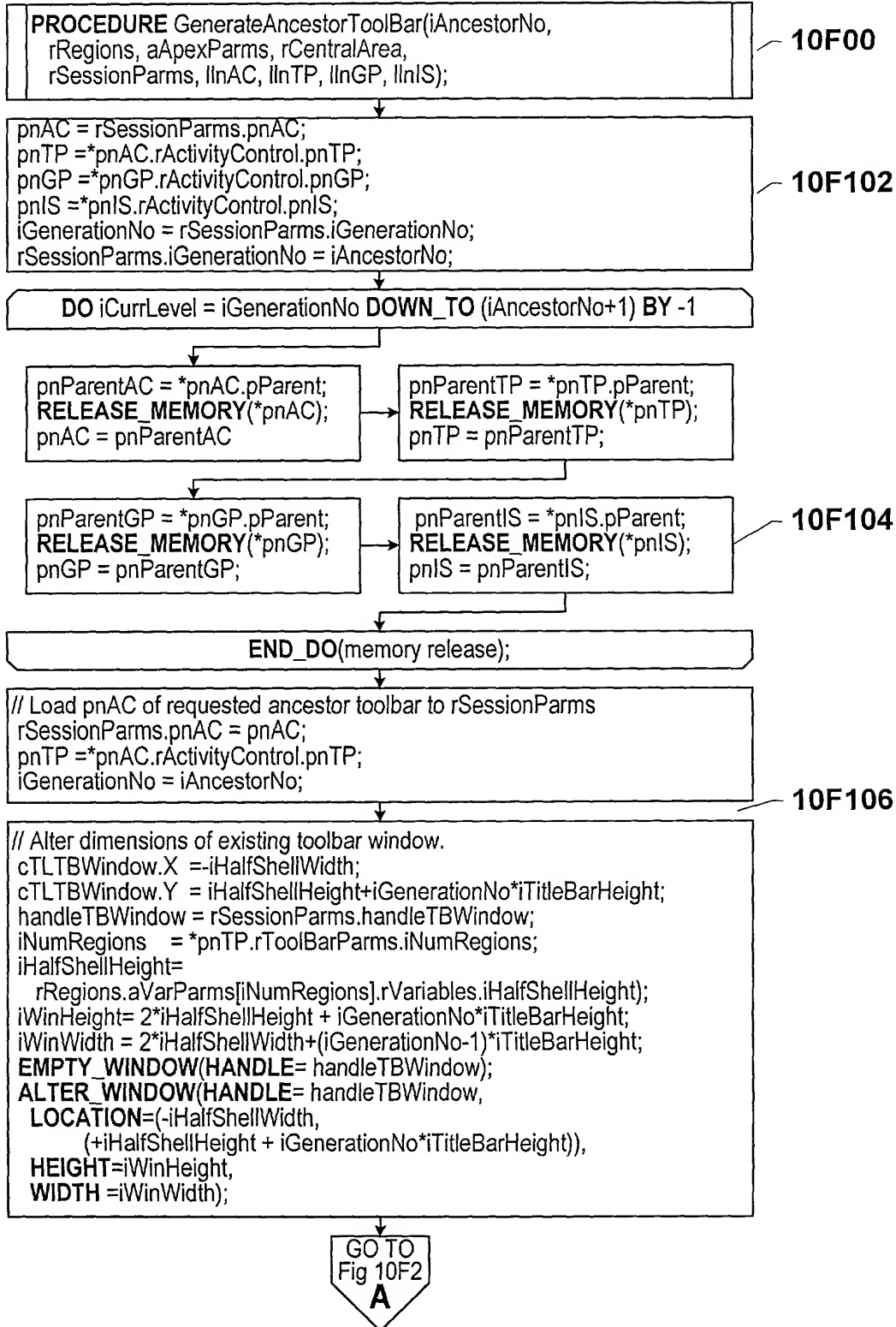
RETURN

FIGURE 10D

FIGURE 10C



66/68



67/68

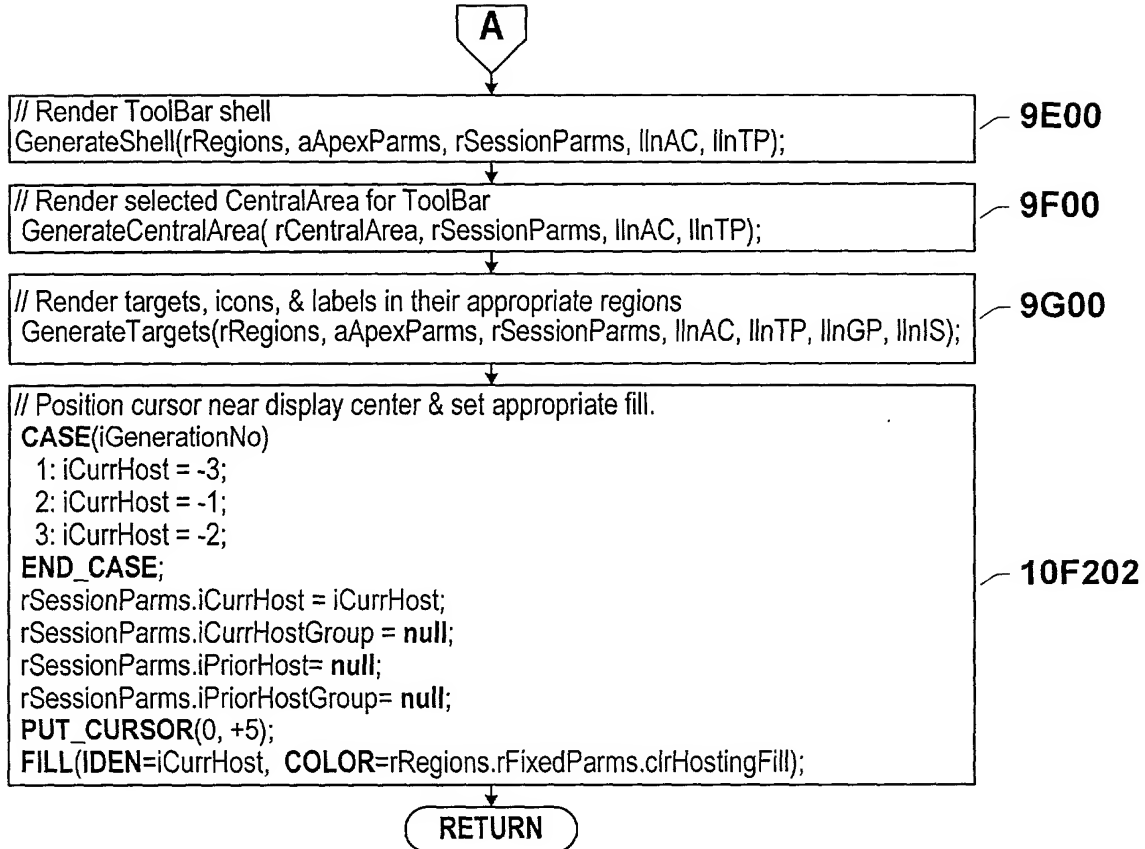


FIGURE 10F2

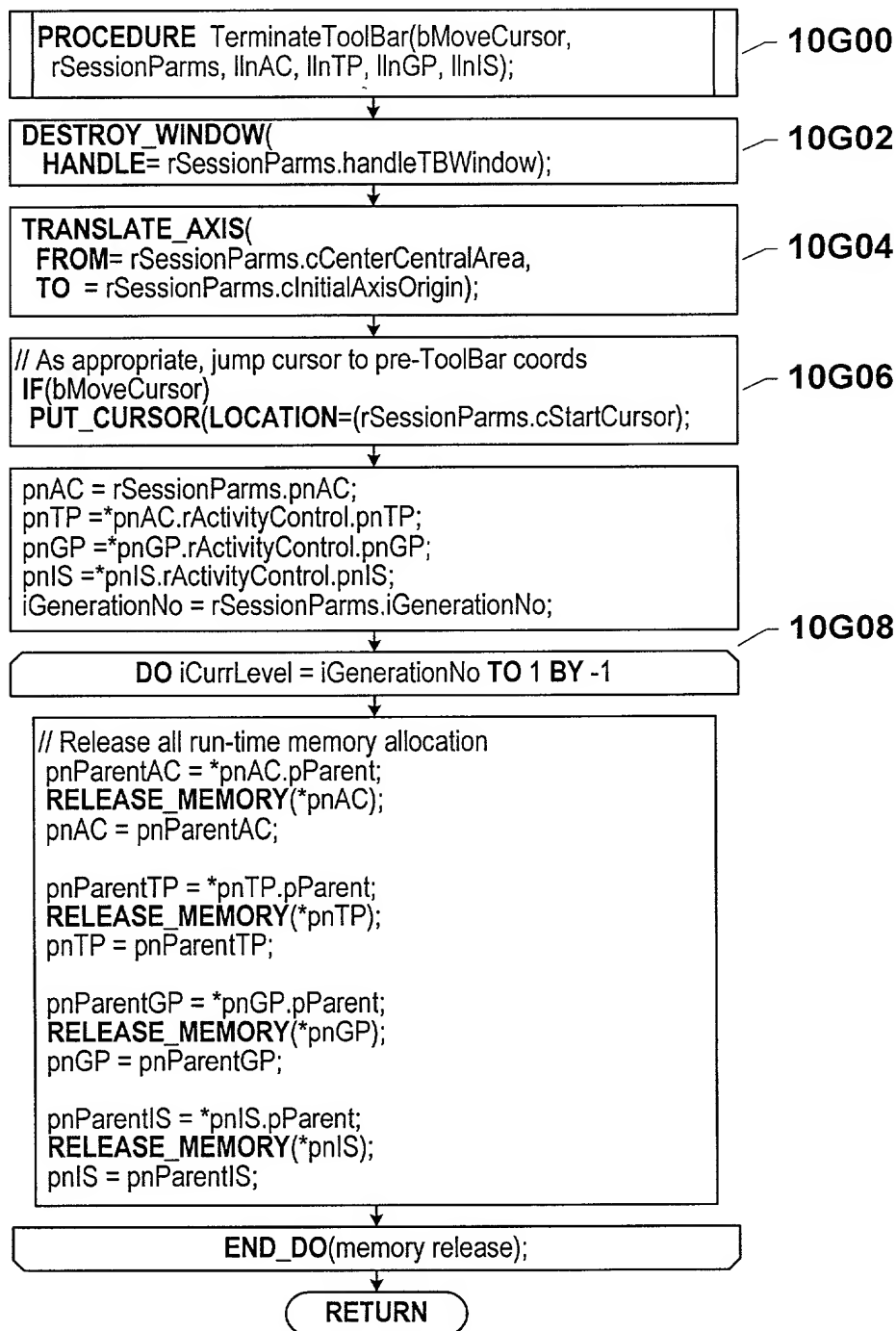


FIGURE 10G